User Manual Converting Library

applications & TOOLS

User Documentation Converting Library
V2.0.0

SIOS ID: 48805235

SIEMENS

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

We reserve the right to make technical changes to this product.

Copyright

Reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration or a utility model or design, are reserved.

User Manual

General Notes

Note

The standard applications are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The standard applications do not represent customer-specific solutions. They are only intended to pro-vide support for typical applications. You are responsible in ensuring that the de-scribed products are correctly used. These standard applications do not relieve you of the responsibility in safely and professionally using, installing, operating and servicing equipment. When using these standard applications, you recognize that Siemens cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these standard applications at any time without prior notice. If there are any deviations between the recommendations provided in these standard applications and other Siemens publications - e.g. Catalogs - then the contents of the other documents have priority.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Warranty, liability and support

We do not accept any liability for the information contained in this document.

Any claims against us - based on whatever legal reason - resulting from the use of the examples, information, programs, engineering and performance data etc., described in this standard application shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract (“wesentliche Vertragspflichten”). However, claims arising from a breach of a condition which goes to the root of the contract shall be limited to the foreseeable damage which is intrinsic to the contract, unless caused by intent or gross negligence or based on mandatory liability for injury of life, body or health The above provisions does not imply a change in the burden of proof to your detriment.

Copyright© 2015 Siemens I DT. It is not permissible to transfer or copy these standard applications or excerpts of them without first having prior authorization from Siemens I DT in writing.

For questions regarding this application please contact us at the following e-mail address:

mailto:applications.erlf.aud@siemens.com

Applicable conditions

/ Alternative 1: (internal business)

If nothing else has been negotiated, then the "General Conditions for the Supply of Products and Services in Siemens Internal Business", valid at the time of sale, are applicable.

/ Alternative 2: (domestic business of Siemens AG)

If nothing else has been negotiated, then the "General License Conditions for Software for Automation and Drives for Customers with a Seat or Registered Office in Germany", valid at the time of sale, are applicable.

/ Alternative 3: (direct export business of Siemens AG)

If nothing else has been negotiated, then the "General License Conditions for Software Products for Automation and Drives for Customers with a Seat or Registered Office outside Germany", valid at the time of sale, are applicable.

/ Alternative 4: (conditions of the particular regional office for the regional office business)

If nothing else has been negotiated, then the "General License Conditions for Software Products for Automation and Drives for Customers with a Seat or Registered Office outside of Germany ", valid at the time of sale, are applicable.

Qualified personnel

In the sense of this documentation qualified personnel are those who are knowledgeable and qualified to mount/install, commission, operate and service/maintain the products which are to be used. He or she must have the appropriate qualifications to carry-out these activities

e.g.:

- Trained and authorized to energize and de-energize, ground and tag circuits and equipment according to applicable safety standards.
- Trained or instructed according to the latest safety standards in the care and use of the appropriate safety equipment.
- Trained in rendering first aid.

There is no explicit warning information in this documentation. However, reference is made to warning information and instructions in the Operating Instructions for the particular product.

Reference regarding export codes

AL: N

ECCN: N

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Table of Contents

Table of Contents 5

Application description 7

1 Introduction 7
1.1 Functional scope 8
1.2 Possible applications 10
1.3 Hardware 10
1.4 Software 10
1.5 Data types, Enumeration types and structures 11
1.6 Abbreviations of the units 11

2 Tools 12
2.1 FCTaskIdentification – Determining the cycle time 12
2.2 Scaling functions 14
2.2.1 FCScalingDWDRL 14
2.2.2 FCScalingRLDWD 15
2.2.3 FCScalingRLWD 16
2.2.4 FCScalingWDRL 17
2.2.5 FCScaling 18
2.3 FCLinear – Linear interpolation 19
2.4 FBPolygonA / FBPolygonB – Non-linear characteristic 21
2.5 FBRFG – Basic ramp-function generator 24
2.6 FBRFGJerkLimit – Ramp-function generator with rounding-off 31
2.7 FCCalculateDistance 39
2.8 FCEvaluateModulo 43

3 Drive functions 46
3.1 FBFrictionMeasurement – Friction characteristic 46
3.2 FBEnableInterface – Activate Interfaces 49

4 Closed-loop control functions 54
4.1 FBPT1 – PT1 element 54
4.2 FBINT – Integrator 58
4.3 FBDIF – Differentiating element 62
4.4 FBDT1 – DT1 filter 64
4.5 FBLIM – Limiter with feedback signal 67
4.6 FBLevelControl – Limit value monitor 69
4.7 FCDeadBand – Dead band element 71
4.8 FCDeadZone – Dead zone element 73
4.9 FBAverageValue – Average value generator 75
4.10 FBPIDController 77
4.11 FBTechnologyController – Technology controller 85

5 Winding functions 93
5.1 FBDiameterCalculation – Diameter calculator 93
5.2 FBTorquePrecontrol – Torque pre-control 114
5.3 FBTaperCharacteristic – Tension taper characteristic 122
5.4 FCRLC – Residual length calculation 127
5.5 FBCalcTargetDiamStopCondition 129
5.6 FCRDC – Target diameter calculation 132
5.7 FBThicknessAverage – High resolution thickness calculation 134
5.8 FBSetUnits – Setting conversion factors 138

Appendix 141

6 General information on the application 141

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

6.1 Scope of supply 141
6.2 Changes/author 141
7 Glossary 142
8 References 142
9 Contact partner 143

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Application description

1 Introduction

The "Converting Library" is intended to provide SIMOTION users with help when engineering and programming converting machinery.

The Converter Library provides standards that can be adapted depending on the particular machine and application. These standards cover the typical basic functions of a converting machine. The functions are available as open source.

The appropriate interfaces are provided to make adaptations depending on the specific machine and application. These interfaces are described in detail in the following documentation. Changes to standard functions that go beyond this should only be made in exceptional cases as this is not in-line with the level of standardization that we wish to achieve here.

Only functions that are not covered by the standard applications are described in this documentation. Additional information is provided in the corresponding application examples.

Additional software packages:

- SIMOTION Rotary Knife
- SIMOTION Flying Saw
- SIMOTION Traversing Drive
- SIMOTION Splice Control
- SIMOTION Winder
- SIMOTION Web Accumulator
- SIMOTION Easy Basics
- SIMOTION Messagehandling
- SIMOTION Axis Function Block

All of the software packages are available through the application database of the Application Centers.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

1.1 Functional scope

Table 1-1

Function	Library / Source	Description
FCTaskIdentification	LConlib / fConvTools	Cycle time identification
FCScalingDWDRL	LConlib / fConvTools	Converting and scaling function to convert a DWORD value into an REAL format.
FCScalingRLDWD	LConlib / fConvTools	Converting and scaling function to convert an REAL value into a DWORD format.
FCScalingWDRL	LConlib / fConvTools	Converting and scaling function to convert a WORD value into an REAL format.
FCScalingRLWD	LConlib / fConvTools	Converting and scaling function to convert an REAL value into a WORD format.
FCScaling	LConlib / fConvTools	Scaling function.
FCLinear	LConlib / fConvTools	Linear interpolation between two points
FBPolygonA	LConlib / fConvTools	Characteristic block Type A
FBPolygonB	LConlib / fConvTools	Characteristic block Type B
FBRFG	LConlib / fConvTools	Basic ramp-function generator
FBRFGJerkLimit	LConlib / fConvTools	Ramp-function generator with jerk limiting
FCCalculateDistance	LConlib / fConvTools	Calculation of breaking distance and time
FCEvaluateModulo	LConlib / fConvTools	Calculation of the position difference between two sampling values of a modulo axis
FBFrictionMeasurement	LConlib / fConvTools	Function to plot a motor friction characteristic.
FBEnableInterface	LConlib / fConvTools	Axis interface activation
FBPT1	LConlib / fConvTools	PT1 filter
FBINT	LConlib / fConvTools	Integrating element
FBDIF	LConlib / fConvTools	Differentiating element
FBDT1	LConlib / fConvTools	DT1 filter
FBLIM	LConlib / fConvTools	Limit value monitor with feedback

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Function	Library / Source	Description
FBLevelControl	LConlib / fConvTools	Limit value monitor with parameterizable limits and hysteresis
FCDeadBand	LConlib / fConvTools	Dead band element
FCDeadZone	LConlib / fConvTools	Dead zone element
FBAverageValue	LConlib / fConvTools	Floating average value generator
FBPidController	LConlib / fConvTools	PID-controller
FBTechnologyController	LconLib / fConvTools	Technology controller
FBWinder	LConlib / fWinder	Center winder function block
FBDiameterCalculation	LConlib / fWinderTools	Diameter calculator for axial winder
FBTorquePrecontrol	LConlib / fWinderTools	Calculating the moment of inertia and the torque pre-control for an axial winder
FBTaperCharacteristic	LConlib / fWinderTools	Tension taper characteristic for the tension setpoint
FCRLC	LConlib / fWinderTools	Calculating the material length on a roll
FCRDC	LConLib /fWinderTools	Calculating the target diameter of a roll.
FBThicknessAverage	LConLib /fWinderTools	Calculation of the material thickness with increased resolution – external option block for FBWinder / FBDiameterCalculation
FBSetUnits	LConlib / fWinderTools	Sets the units for the winder blocks.
Limit, Min, Max	<System function>	Limiting, Minimum, Maximum determination
_getAxisStoppingData	<System function>	Function to calculate the braking distance and the braking time when specifying the velocity, acceleration and jerk at the start of braking.
_readDriveParameter _readDriveMultiParameter _writeDriveParameter _writeDriveMultiParameter	<System function>	Function to read/write drive parameters.
_readDriveFaults	<System function>	Reading the actual fault buffer entries of the drive

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

1.2 Possible applications

The Converting Library can be used for the widest range of applications, e.g.:

- Foil machines
- Printing machines
- Coating machines
- Paper finishing machines
- Wire drawing machines
- Textile machines
- Post printing machines

1.3 Hardware

The functions of the Converting Library are available for the following hardware platforms:

- SIMOTION C2xx
- SIMOTION P3xx
- SIMOTION D4xx

1.4 Software

The following software versions are required to implement the standard functions contained in this library:

- SIMOTION SCOUT from V4.2
- SIMOTION kernel from V4.2
- SIMOTION technology packages from V4.2

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

1.5 Data types, Enumeration types and structures

The interface parameters of each function or function block are described in this document in the respective sections.

Additionally to the interface parameter, some of the function blocks using a configuration structure for parameterization and a diagnostic structure for feedback reason. These structures are defined as sub-structures of the sWinderConfig and sWinderDiag structures and are described in the FBWinder documentation.

1.6 Abbreviations of the units

The abbreviations of the units used for the speed and positioning axes are shown in the following table in order to ensure consistency for the metric as well as the Anglo-American dimension system.

LU stands for a length unit, TU stands for a time unit.

Table 1-2: Abbreviations of the units

Measurement quantity	Units	Metric	Anglo-American
Length	LU	e.g.: m (Meter)	e.g.: ft (feet)
Length (basis unit)	LUb	m (meters)	ft (feet)
Velocity	LU / TU	e.g.: mm / min	e.g.: inch / min
Acceleration	LU / TU²	e.g.: m / s²	e.g.: ft / s²
Speed	rpTU o. ° / TU	e.g.: rev / min	e.g.: rpm
Mass, weight	m	kg	lb
Density	m / LUb³	kg / m³	lb / ft³
Moment of inertia	J = mLUb²	kgm²	lbft²
Torque	M = mLUb²/s²	Nm = kgm² / s²	lbf ft = lbft² / s²
Tension, force	Z = mLUb/s²	N = kg m / s²	lbf = lb ft / s²

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

2 Tools

2.1 FCTaskIdentification – Determining the cycle time

Identifies the task and outputs the associated cycle time

Block name

FCTaskIdentification

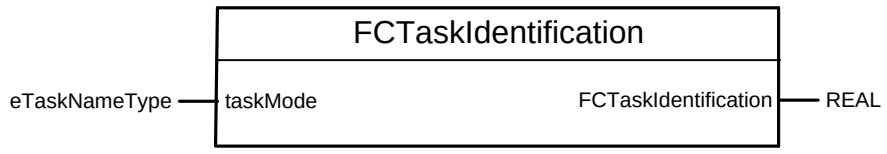
Task

The function identifies the type of task in which this is called, determines the actual cycle time and outputs this.

Integration in the run-time system

The function must be called in a synchronous task (Servo, IPO, IPO_2 or Timer).

Graphic representation of the block



Block parameters

Input parameters

Name	Data type	Initial value	Description
taskEnum	eTaskName Type	-	Calling task: SERVO_SYNCHRONOUS_TASK: Servo Synchronous Task IPO_SYNCHRONOUS_TASK: Ipo Synchronous Task IPO_SYNCHRONOUS_TASK_2: Ipo Synchronous Task 2 TIMER_INTERRUPT_TASK: in a Timer Interrupt Task

Output parameters

Name	Data type	Initial value	Description
FCTaskIdentification	REAL	-	[ms] The actual real cycle time of the task in the REAL format, If the function outputs a 0, the task is not time synchronous and Ipo-dependent.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Functionality

Using the input parameter *taskMode* of type eTaskNameType, the function determines the actual task in which the FC is called. If the task is known, then the cycle time can be determined. By mapping the cycle time in the Time Format in the TaskStartInfo, this time must be interrogated using the system variables and precisely calculated. The precise cycle time of the task is output at *FCTaskIdentification* output in the REAL format. This function can only work when called in a time synchronous Ipo-dependent task. 0.0 is output when called in another task.

In systems where the cycle time of the IPO task shorter is than 1.0ms the cycle time of the TIMER_INTERRUPT_TASK can not be identified. In this case the function returns with 0.0.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

2.2 Scaling functions

2.2.1 FCScalingDWDRL

Block name

FCScalingDWDRL

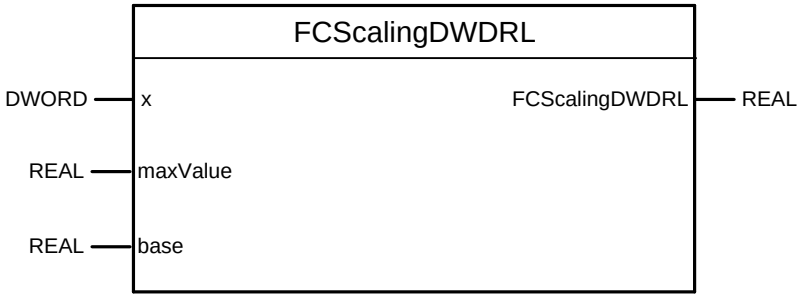
Task

Converting a DWORD into an REAL value

Integration in the run-time system

The function can be called in any task.

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Block parameters

Input parameters

Name	Data type	Initial value	Description
x	DWORD	-	[-] Input value as DWORD
maxValue	REAL	2147483647.0	[-] Maximum value
base	REAL	-	[-] Normalization value (*FCScalingDWDRL* = 0 if *base* = 0)

Output parameters

Name	Data type	Initial value	Description
FCScalingDWDRL	REAL	-	[-] Output value in REAL [-2*maxValue - 2*maxValue]

Functionality

The function converts a DWord (*x*) using a normalization value (*base*) and the maximum reference value (*maxValue*) into a real value (*FCScalingDWDRL*). In so doing, the output value is limited between the values *-2*maxValue* and *2*MaxValue*.

$$FCScalingDWDRL = \frac{x}{base} * \max Value$$

2.2.2 FCScalingRLDWD

Block name

FCScalingRLDWD

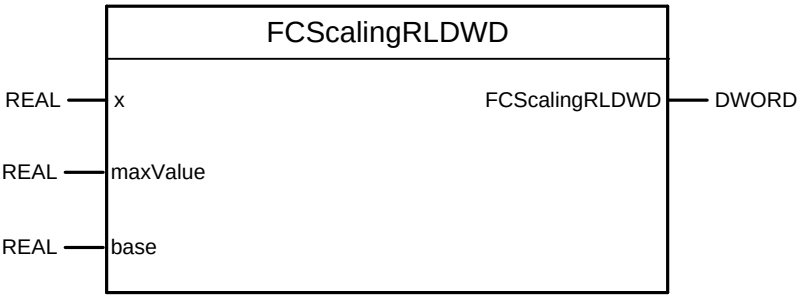
Task

The block converts a REAL value into a DWORD

Integration in the run-time system

The function can be called in any task.

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Block parameters

Input parameters

Name	Data type	Initial value	Description
x	REAL	-	[-] Input value as REAL [-2*maxValue - 2*maxValue]
maxValue	REAL	-	[-] Maximum value (*FCScalingRLDWD* = 0 if *maxValue* = 0)
base	REAL	2147483647.0	[-] Normalization value

Output parameters

Name	Data type	Initial value	Description
FCScalingRLDWD	DWORD		[-] Output value as DWORD

Functionality

The function converts a real value (*x*) into a DWord (*FCScalingLRLDWD*) using a normalization value (*base*) and the maximum reference value (*maxValue*). In so doing, input value *x* is limited between the values *-2*maxValue* and *2*maxValue*.

$$FCScalingRLDWD = \frac{x}{\max Value} * base$$

2.2.3 FCScalingRLWD

Block name

FCScalingRLWD

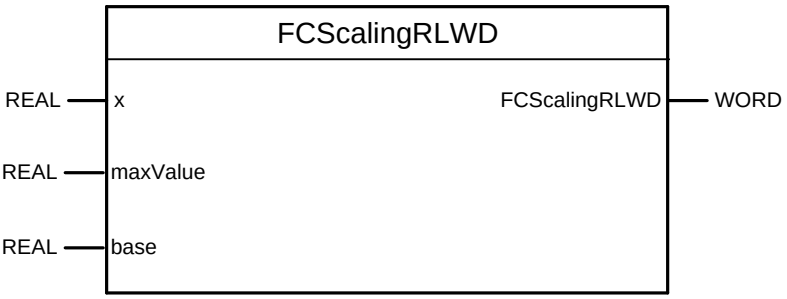
Task

This block converts a REAL value into a WORD value

Integration in the run-time system

The function can be called in any task.

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Block parameters

Input parameters

Name	Data type	Initial value	Description
x	REAL	-	[-] Input value as REAL [-2*maxValue - 2*maxValue]
maxValue	REAL	-	[-] Maximum value (*FCScalingRLWD* = 0 if *maxValue* = 0)
base	REAL	16384.0	[-] Normalization value

Output parameters

Name	Data type	Initial value	Description
FCScalingRLWD	WORD	-	[-] Output value as WORD

Functionality

This function converts a real value (*x*) into a word value (*FCScalingRLWD*) using a normalization value (*base*) and the maximum reference value (*maxValue*). In so doing, input value *x* is limited between the values *-2*maxValue* and *2*maxValue*.

$$FCScalingRLWD = \frac{x}{\max Value} * base$$

2.2.4 FCScalingWDRL

Block name

FCScalingWDRL

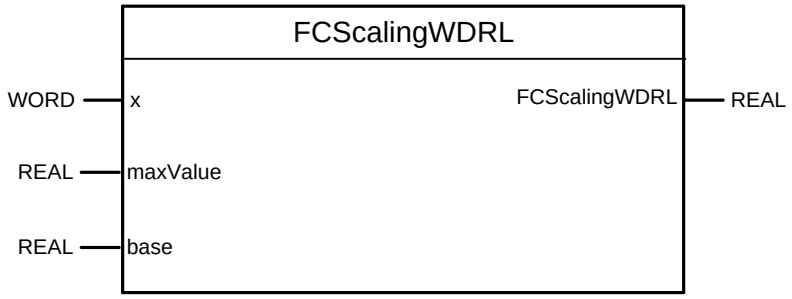
Task

This block converts a WORD value into a REAL value

Integration in the run-time system

The function can be called in any task.

Graphic representation of the block



Block parameters

Input parameters

Name	Data type	Initial value	Description
x	WORD	-	[-] Input value as WORD
maxValue	REAL	-	[-] Maximum value
base	REAL	16384.0	[-] Normalization value (*FCScalingWDRL* = 0 if *base* = 0)

Output parameters

Name	Data type	Initial value	Description
FCScalingWDRL	REAL	-	[-] Output value in REAL [-2*maxValue - 2*maxValue]

Functionality

This function converts a word value (*x*) into a Real value (*FCScalingWDRL*) using a normalization value (*base*) and the maximum reference value (*maxValue*). In so doing, the output value is limited between the values *-2*maxValue* and *2*maxValue*.

$$FCScalingWDRL = \frac{x}{base} * \max Value$$

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

2.2.5 FCScaling

Block name

FCScaling

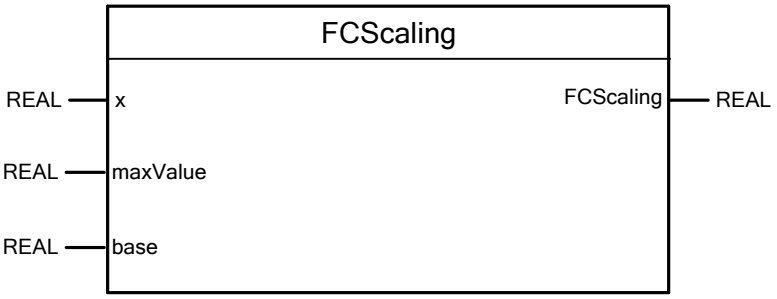
Task

This block scales a real value

Integration in the run-time system

The function can be called in any task.

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Block parameters

Input parameters

Name	Data type	Initial value	Description
x	REAL	0.0	[-] Input value as WORD
maxValue	REAL	0.0	[-] Maximum value
base	REAL	16384.0	[-] Normalization value (*FCScaling* = 0 if *base* = 0)

Output parameters

Name	Data type	Initial value	Description
FCScaling	REAL	-	[-] Output value in REAL

Functionality

This function scales a word value (*x*) using a normalization value (*base*) and the maximum reference value (*maxValue*).

$$FCScaling = \frac{x}{base} * \max Value$$

2.3 FCLinear – Linear interpolation

Block name

FCLinear

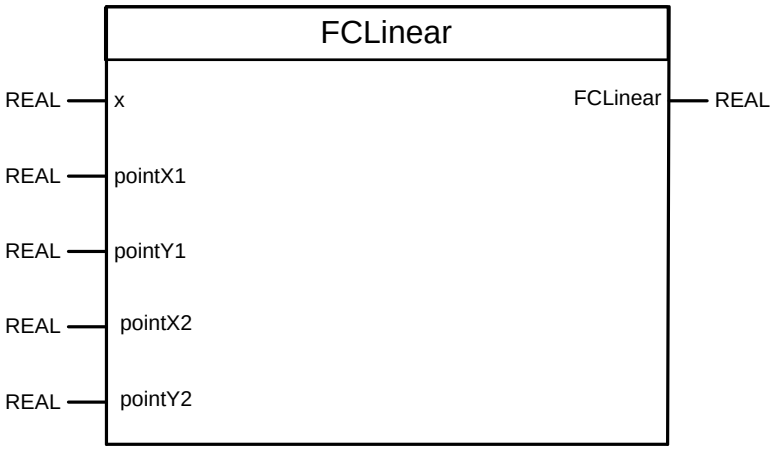
Task

This function interpolates a point from a specified straight line.

Integration in the run-time system

The function can be called in any task.

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Block parameters

Input parameters

Name	Data type	Initial value	Description
x	REAL	0.0	Input point X
pointX1	REAL	0.0	X Coordinate point 1
pointY1	REAL	0.0	Y Coordinate point 1
pointX2	REAL	0.0	X Coordinate point 2
pointY2	REAL	0.0	Y Coordinate point 2

Output parameters

Name	Data type	Initial value	Description
FCLinear	REAL	-	Output point Y

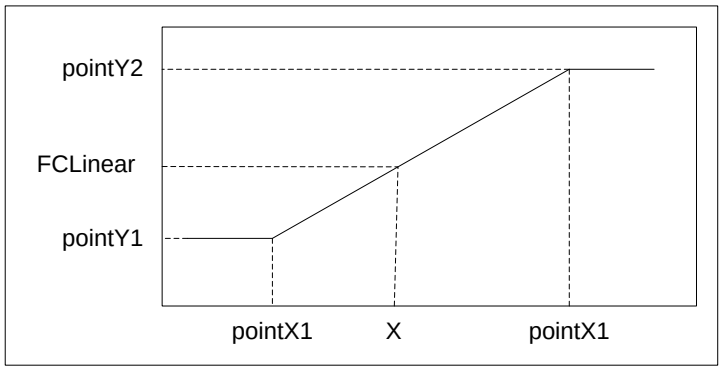
Functionality

Using the straight lines entered, this function determines the matching Y coordinate of the coordinate X present at input *x*. The straight line is defined by point 1 *(pointX1 / pointY1)* and point 2 *(pointX2 / pointY2)*. The interpolated Y coordinate is provided as return value of the function with *FCLinear*. The input point *x* is limited by the coordinates pointX1 and pointX2. The calculation is continued with *pointX1* or *pointX2*. If the coordinates *pointX1* and *pointX2* are equal, 0.0 is displayed at output *FCLinear* and no interpolation is carried-out.

$$FCLinear = \frac{X - point X1}{point X2 - point X1} * (point Y2 - point Y1) + point Y1$$

Graphic representation of the function

Figure 2-1: Graphic representation of the function



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

2.4 FBPolygonA / FBPolygonB – Non-linear characteristic

This block determines a value from a non-linear characteristic that is entered.

Block name

FBPolygonA, FBPolygonB

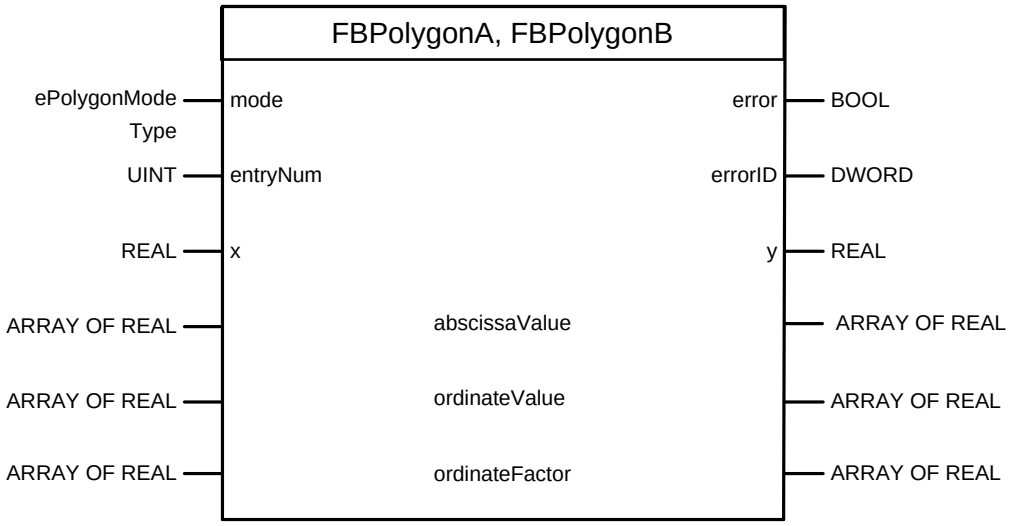
Task

The function block implements the function of a non-linear characteristic by linearly interpolating between two table values entered using the input quantities.

Integration in the run-time system

The function block can be called in any task.

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Block parameters

Input parameters

Name	Data type	Initial value	Description
mode	ePolygonMode Type	ORDINATE_ FACTORS_ NEGLECTED	The individual weighting is enabled NO_POLYGON: No Polygon calculation (y = x). ORDINATE_FACTORS_USED: Weighting successful, interpolated with the product (ordinateValue*ordinateFactor). ORDINATE_FACTORS_NEGLECTED: No weighting, interpolated with (ordinateValue).
entryNum	UINT	0	FBPolygonA Number of grid points [1… LCONLIB_ LCONLIB_MAX_IDX_A+1] (21) FBPolygonB Number of grid points [1 … LCONLIB_ LCONLIB_MAX_IDX_B+1] (181)

Converting Library

Name	Data type	Initial value	Description
x	REAL	0.0	[-] Actual input value of the FB

Output parameters

Name	Data type	Initial value	Description
error	BOOL	FALSE	FALSE: No error TRUE: An error has occurred, refer to error identification
errorID	DWORD	16#0000_0000	Error identification, refer to error messages errorID <= 16#000_8000: warning errorID >= 16#0000_8000: error
y	REAL	0.0	[-] Interpolated output value of the FB

Input/Output parameters

Name	Data type	Initial value	Description
abscissaValue	ARRAY [0.. LCONLIB_MAX_IDX_x] OF REAL	-	FBPolygonA: Abscissa grid points, monotonically (steadily) increasing from a value perspective [2 … LCONLIB_MAX_IDX_A+1] FBPolygonB: Abscissa grid points, monotonically (steadily) increasing from a value perspective [2 … LCONLIB_MAX_IDX_B+1]
ordinateValue	ARRAY [0.. LCONLIB_MAX_IDX_x] OF REAL	-	FBPolygonA: Ordinate grid points [2 … LCONLIB_MAX_IDX_A+1] FBPolygonB: Ordinate grid points [2 … LCONLIB_MAX_IDX_B+1]
ordinateFactor	ARRAY [0.. LCONLIB_MAX_IDX_x] OF REAL	-	FBPolygonA: Ordinate weighting factor [2 … LCONLIB_MAX_IDX_A+1] FBPolygonB: Ordinate weighting factor [2 … LCONLIB_MAX_IDX_B+1]

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Functionality

Input *mode* specifies a mode for the function block to calculate the interpolating value *y*; this is determined using the input *x*. For the mode NO_POLYGON there is no calculation and the Output Value is the same like the Input Value. In mode ORDINATE_FACTORS_USED, the *factors* in the field *ordinateFactors* are included in the calculation *(ordinateValue * ordinateFactor)*.

For the mode ORDINATE_FACTORS_NEGLECTED, no weighting is applied and only the values from *ordinateValue* are used. A maximum of either 21 (default FBPolygonA) or 181 (default FBPolygonB) abscissa grid points are evaluated that are specified to be monotonically (steadily) increasing from a value perspective using a field:

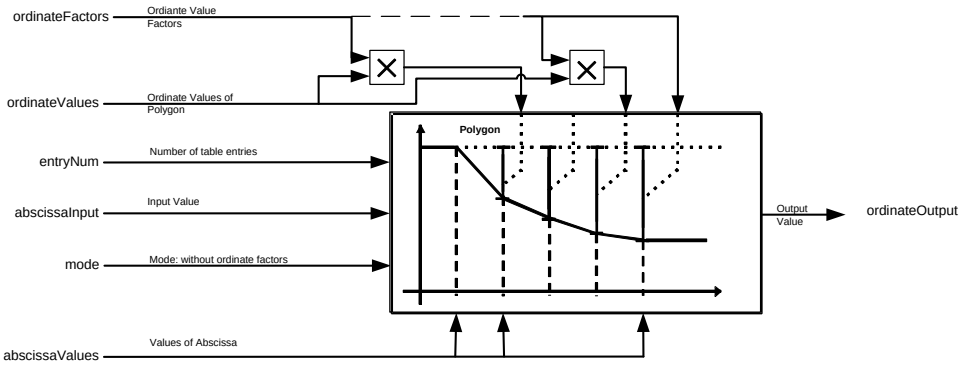
abscissaValue[i] <= abscissaValue[i+1], otherwise, an error is output and *ordinateOutput* keeps the last value.

The values of the two arrays *ordinateValue* and *ordinateFactor* correspond to the abscissa via indices (0<=i<=LCONLIB_MAX_IDX_x+1)

This block is available in two versions, FBPolygonA with 21 grid points and FBPolygonB with 181 grid points.

Graphic representation of the function

Figure 2-2: Block diagram



Warnings

Warnings are indicated by the status of the outputs *error* and *errorID*: *error* = FALSE and *errorID* <> 16#0000_0000.

ErrorID	Description
16#0000_0000	No warning
16#0105_4005	The table of entered values is not monotonically (steadily) increasing -> no matching value found
16#0105_4008	The table of entered values is not monotonically (steadily) increasing -> 2 values are identical

Errors

Error messages are indicated by the status of the outputs *error* and *errorID*: *error* = TRUE and *errorID* <> 16#0000_0000.

ErrorID	Description
16#0105_8003	The number of grid points that have been entered is too small
16#0105_8004	The number of grid points that have been entered is too large

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

2.5 FBRFG – Basic ramp-function generator

Basic ramp-function generator without rounding-off

Block name

FBRFG

Task

The basic ramp-function generator is a ramp-function generator to limit the rate of change of input quantity x. The output quantity of the FBRFG can be limited.

The following quantities can be independently set and changed during operation:

- Ramp-up and ramp-down times
- Output limiting upper Limit and lower Limit
- Setting value

Flexible ramp-function generator function:

- Integrating, tracking setpoint x
- The ramp-function generator output can be set to the initial value (load setValue into the supply integrator)
- The ramp-function generator output can be increased and decreased, integrating

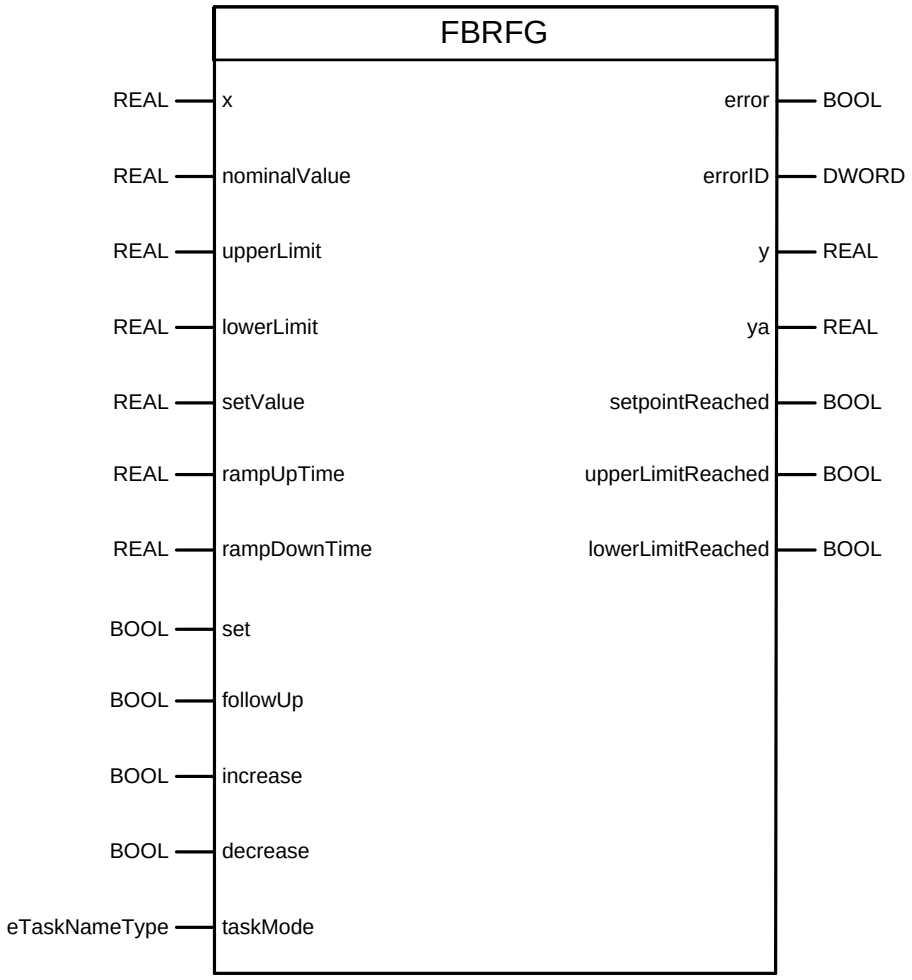
Integration in the run-time system

The function block must be called in a synchronous task (Servo, IPO, IPO_2 or Timer).

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Block parameters

Input parameters

Name	Data type	Initial value	Description
x	REAL	0.0	[-] Input quantity
nominalValue	REAL	1.0	[-] Normalization value
upperLimit	REAL	0.0	[-] Upper limit value
lowerLimit	REAL	0.0	[-] Lower limit value
setValue	REAL	0.0	[-] Setting value
rampUpTime	REAL	0.0	[ms] Ramp-up time [cycle time – MaxTU]
rampDownTime	REAL	0.0	[ms] Braking time constant [cycle time – MaxTD]
set	BOOL	FALSE	FALSE: A setting value is not accepted/transferred TRUE: A setting value is accepted/transferred
followup	BOOL	FALSE	FALSE: Normal sequence TRUE: Output is set the same as the input

Converting Library

Name	Data type	Initial value	Description
increase	BOOL	FALSE	FALSE: Normal sequence TRUE: Higher
decrease	BOOL	FALSE	FALSE: Normal sequence TRUE: Lower
taskMode	eTaskName Type	IPO_ SYNCHRONOUS_ TASK	Specification of the task in which the block is called. SERVO_SYNCHRONOUS_TASK IPO_SYNCHRONOUS_TASK IPO_SYNCHRONOUS_TASK_2 TIMER_INTERRUPT_TASK

Output parameters

Name	Data type	Initial value	Description
error	BOOL	FALSE	FALSE: No error TRUE: An error has occurred, refer to error identification
errorID	DWORD	0	Error identification, refer to error messages errorID <= 16#000_8000: warning errorID >= 16#0000_8000: error
Y	REAL	0.0	[-] Output quantity
ya	REAL	0.0	[- / s] Acceleration value
SetpointReached	BOOL	FALSE	FALSE: Setpoint calculation active TRUE: Setpoint (input *x*) reached
upperLimit Reached	BOOL	FALSE	FALSE: Upper limit value not reached TRUE: Upper limit value reached
lowerLimit Reached	BOOL	FALSE	FALSE: Lower limit value not reached TRUE: Lower limit value reached

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Functionality

The function block includes an integrator with two integration time constants that can be set independently of one another. Output *Y* changes according to the following algorithm:

$$Yn = Y_{n-1} + YA_n$$

The acceleration value Y_A is separately calculated for the up ramp and down ramp and is output. An up ramp is when the output value *Y* moves away from zero. A down ramp is when the output value *Y* approaches zero.

For the acceleration value when ramping-up the following applies:

$$YA = \frac{TA}{rampupTime} * no\min alValue \quad für\, Y > 0$$

$$YA = -\frac{TA}{rampupTime} * no\min alValue \quad für\, Y < 0$$

For the acceleration value when ramping-down the following applies:

$$YA = -\frac{TA}{rampdownTime} * no\min alValue \quad für\, Y > 0$$

$$YA = \frac{TA}{rampDownTime} * no\min alValue \quad für\, Y < 0$$

T_A is the sampling time in which the function block is processed.

The changeover between ramp-up time and ramp-down time is realized when the direction changes or at the zero crossover of the transfer function. Depending on the logical states of control inputs *set, followup, increase* and *decrease*, the operating mode is specified by a control logic. The output quantity can be limited using inputs *upperLimit* and *lowerLimit*. When the set limits are reached, through *Y*, binary outputs *upperLimitReached* or *lowerLimitReached* are set to TRUE. Binary output *setpointReached* goes TRUE, if *y=x*.

Ramp-up time and ramp-down times

The ramp-up time *rampupTime* is the time in which the absolute value of the output quantity increases by *nominalValue*.

The ramp-down *rampdownTime* is the time in which the absolute value of the output quantity decreases by *nominalValue*.

The ramp-up time and ramp-down time can be selected to be different. The lower T_A/*rampupTime* or T_A/*rampDownTime*, the lower the amplitude change at *y* from one sampling instant to the next.

For the control inputs, the following sequence of priority applies:
set before *followup* before *increase* and *decrease*

Table 2-1: Function of the control inputs

set = 1	Setting value setValue is loaded into the integrator, non-integrating
followup = 1	Output y integrating to setpoint X, tracking
Increase = 1	Output y integrating torques upperLimit, tracking
decrease = 1	Output y integrating torques lowerLimit, tracking

Operating modes and control of the ramp-function generator

The combination of commands at the control inputs and the possible operating modes can be taken from the truth tables.
Standard ramp-function generator operation involves *lowerLimit* **≤** *0* **≤** *upperLimit* and *lowerLimit* **<** y_n **<** *upperLimit*. However, other settings are also possible that are explained in the following.

For the setting with *lowerLimit* **≥** *upperLimit*, the following applies: Limit *upperLimit* is dominant with respect to limit *lowerLimit*.

Behavior of the integrator at the limit

If output y reaches one of the set limits *lowerLimit* or *upperLimit***,** during the control sequence, the integrator value is held. Output value *y* is then kept constant until the integrator value leaves the limit due to a change of the input quantities.

If the integrator is at the limit and the limit value is changed the integrator behaves differently depending on the direction in which the limit value was changed.

If the absolute value of a limit value is increased, and if the control logic specifies that the ramp-function generator should run in the same direction, then the integrator integrates from the previously defined value according to the selected ramp-up time until the output again reaches the limit value.

If the absolute value of a limit value is reduced, the integrator integrates from the previously held value – according to the selected ramp-down time – until the output again reaches the limit value.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

The integrator operates internally with an increased accuracy so that even for small setpoint-actual value differences it still integrates. However, it should be carefully noted that the sampling time is sufficiently low referred to the ramp-up or ramp-down time.

rampupTime and T_D are internally limited:

rampupTime ≥ T_A, *rampdownTime* ≥ T_A.

Table 2-2: Truth table, ramp-function generator stop

set	follow up	increase	decrease	ya_n	y_n	Mode	Comment
0	0	0	0	0	y_{n-1}	Stop	y is constant
1	1)	1)	1)	Step	$setValue_n$	Sets output to setValue	setValue is any value, fixed or variable
0	1	1)	1)	T_A/rampupTime; T_A/rampdownTime	$y_{n-1}+ya_n$	Normal operation y→ x	rampupTime for [x > y ∧ y ≥ 0] ∨ [x < y ∧ y ≤ 0] rampdownTime for [x > y ∧ y < 0] ∨ [x < y ∧ y > 0]
0	0	1	0	T_A/ rampupTime (T_A/ rampdownTime)	$y_{n-1}+ya_n$	Approach the upper limit value y→upperLimit	rampupTime, rampdownTime as before, dependent on the starting (initial) position
0	0	0	1	T_A/ rampdownTime (T_A/ rampupTime)	$y_{n-1}+ya_n$	Approach the lower limit value y→lowerLimit	rampupTime, rampdownTime as before, dependent on the starting (initial) position

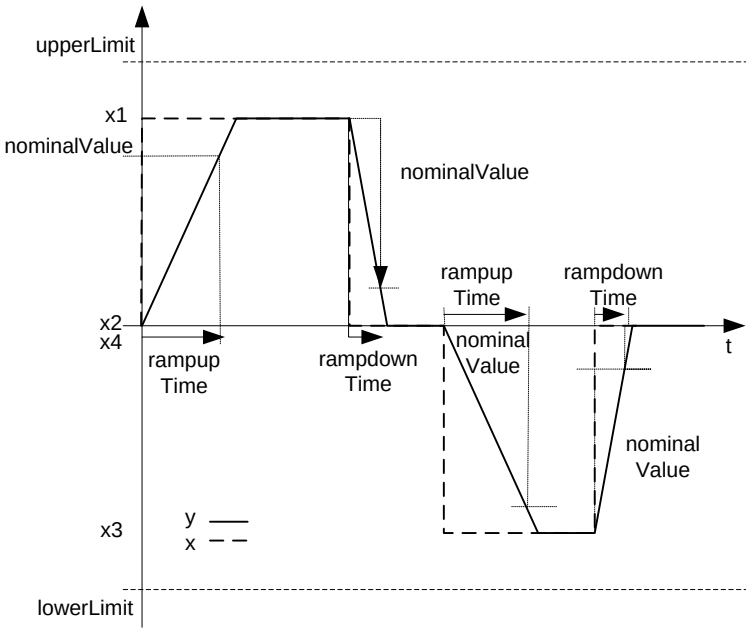
1) = any

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Graphic representation of the function

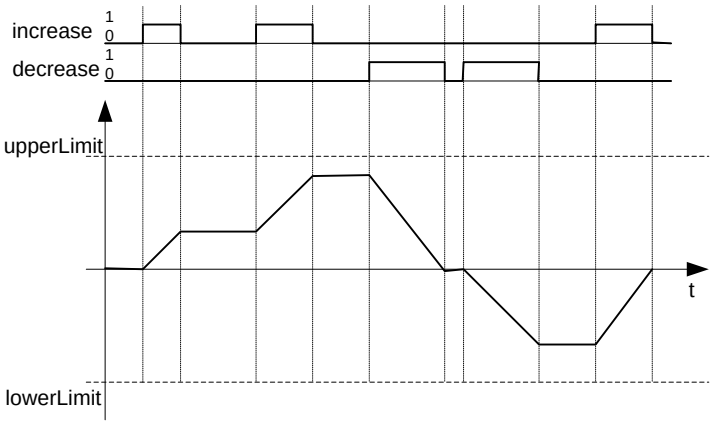
Transfer Function

Figure 2-3: Transfer function



Example 1**:** *followup* = 1 with *lowerLimit* < *upperLimit* and *lowerLimit* < x < *upperLimit*, as well as x1 = 1.5, x2 = x4 = 0.0, x3 = –1.5, *upperLimit* = 2.0, *lowerLimit* = –2.0, *rampupTime* > *rampdownTime*

Figure 2-4

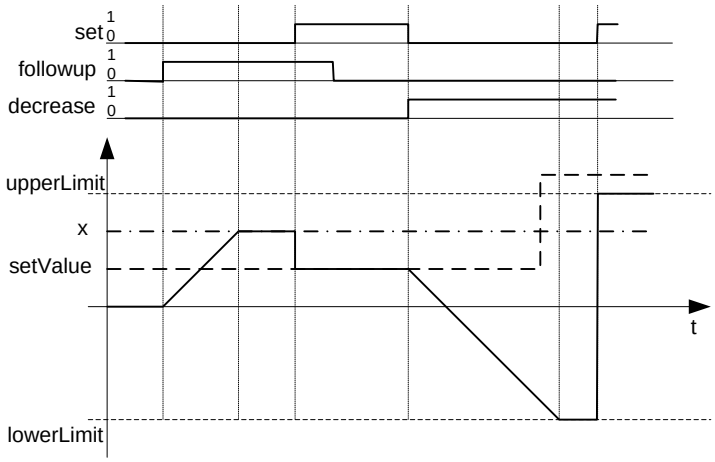


Example 2: Motor potentiometer function with *increase* and *decrease* and with *lowerLimit* < *upperLimit*

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Figure 2-5



Example 3: Setting the integrator with *lowerlimit* < *upperLimit*

Errors

Error messages are indicated by the status of the outputs *error* and *errorID*: *error* = TRUE and *errorID* <> 16#0000_0000.

ErrorID	Description
16#0000_0000	No error.
16#0106_8001	The function block is called in an incorrect task.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

2.6 FBRFGJerkLimit – Ramp-function generator with rounding-off

Ramp-function generator with ramp-up and ramp-down times, that can be set and rounding-off.

Block name

FBRFGJerkLimit

Task

Depending on the entered integration values and the specified setpoint function block FBRFGJerkLimit calculates an appropriate output value. In so doing, the acceleration and deceleration times are taken into account as well as a selected rounding-off. There is a special input for a fast stop. The status of the ramp-function generator is output via additional outputs.

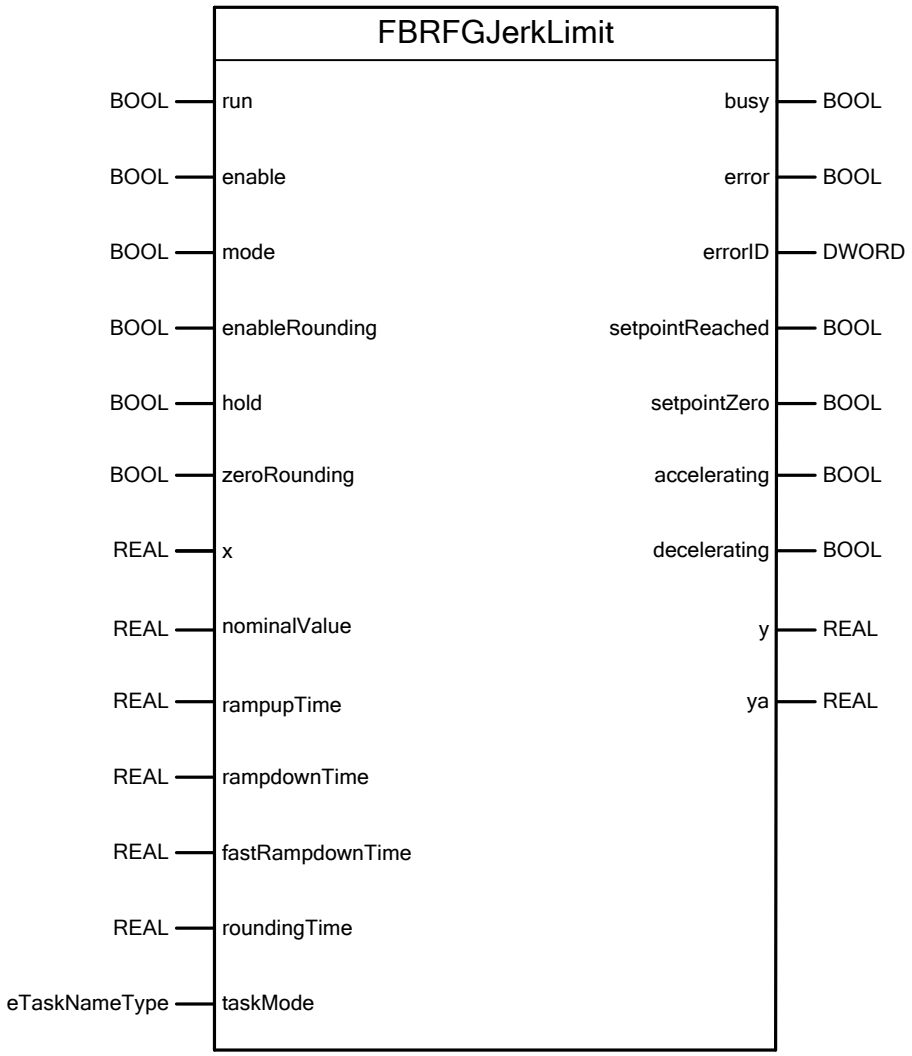
Integration in the run-time system

The function block must be called in a synchronous task (Servo, IPO, IPO_2 or Timer).

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Block parameters

Input parameters

Name	Data type	Initial value	Description
run	BOOL	FALSE	FALSE: FB de-activated TRUE: FB activated
enable	BOOL	FALSE	FALSE: No calculation TRUE: The output values are calculated
mode	BOOL	TRUE	FALSE: Emergency Stop TRUE: Normal sequence
enableRounding	BOOL	TRUE	FALSE: Without rounding-off TRUE: With rounding-off
hold	BOOL	FALSE	TRUE: Ramping will be hold, with respect to jerk limiting. Acceleration will be ramped to zero. The input x is ineffective. FALSE: Normal sequence
zeroRounding	BOOL	FALSE	TRUE: Rounding at zero crossing FALSE: No rounding at zero crossing
x	REAL	0.0	Setpoint, always positive
nominalValue	REAL	1000.0	Nominal Value Internally, the absolute value is always taken from this input value
rampupTime	REAL	1.0	[s] Ramp-up time Internally limited to between 0.001 – 1000.0
rampdownTime	REAL	1.0	[s] Ramp-down time Internally limited to between 0.001 – 1000.0
fastRampdownTime	REAL	0.1	[s] Fast stop ramp-down time Internally limited to between 0.001 – 1000.0
roundingTime	REAL	0.5	[s] Rounding-off time Internally limited to between 0.001 – 1000.0
taskMode	eTaskName Type	IPO_ SYNCHRONOUS_ TASK	Specification of the task in which the block is called. SERVO_SYNCHRONOUS_TASK IPO_SYNCHRONOUS_TASK IPO_SYNCHRONOUS_TASK_2 TIMER_INTERRUPT_TASK

Output parameters

Name	Data type	Initial value	Description
busy	BOOL	FALSE	FALSE: It is not calculated TRUE: FB calculates an Output_Value
error	BOOL	FALSE	FALSE: No error TRUE: An error has occurred, refer to error identification
errorID	DWORD	16#0000_0000	Error identification, refer to error messages errorID <= 16#000_8000: warning errorID >= 16#0000_8000: error
Setpoint Reached	BOOL	FALSE	FALSE: Setpoint not reached TRUE: Setpoint reached
setpointZero	BOOL	FALSE	FALSE: Calculated setpoint is not zero TRUE: Calculated setpoint is zero

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Name	Data type	Initial value	Description
accelerating	BOOL	FALSE	FALSE: Is not accelerated TRUE: Is accelerated
decelerating	BOOL	FALSE	FALSE: Is not decelerated TRUE: Is decelerated
y	REAL	0.0	Calculated output value
ya	REAL	0.0	[1/sec²] Actual acceleration / deceleration

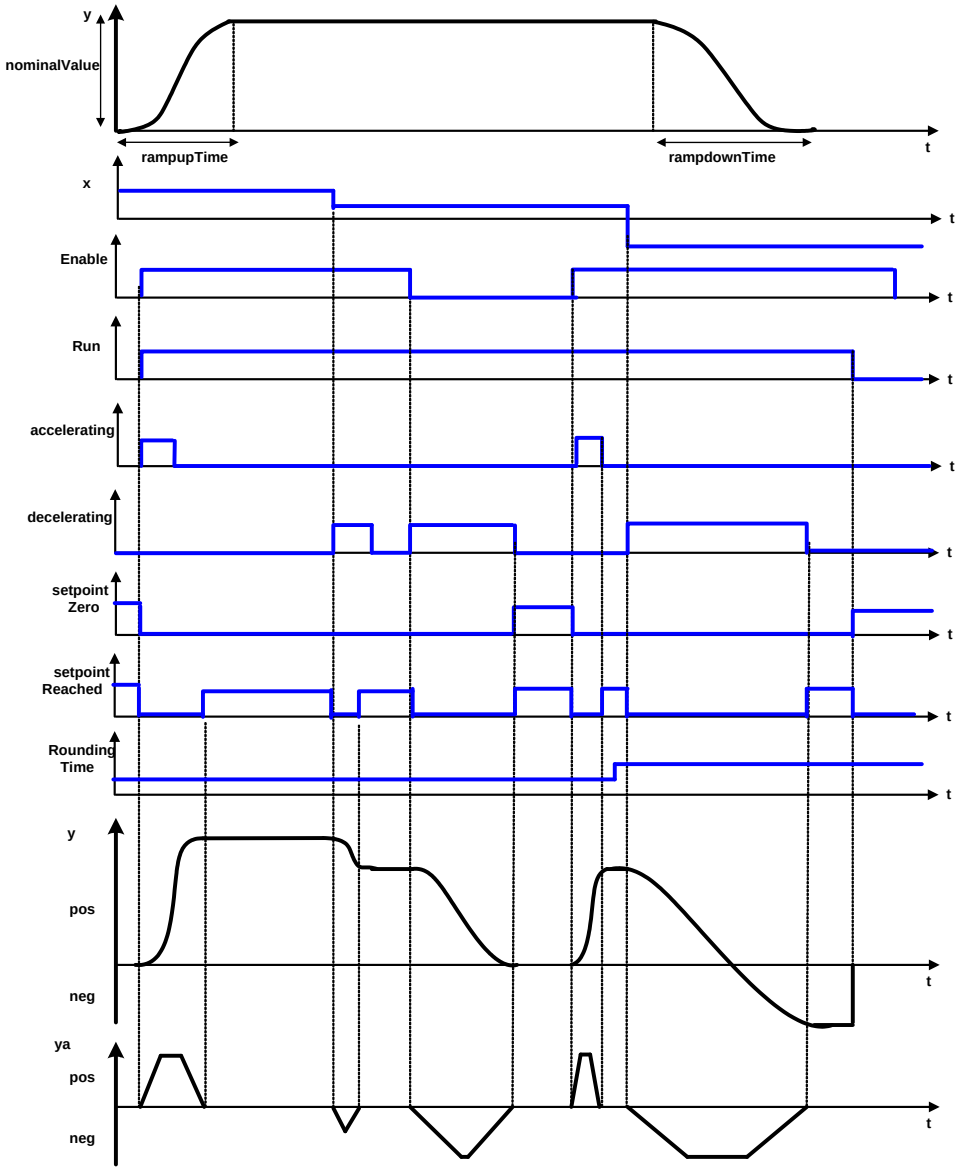
Functionality

The function block is active with the input *run* at TRUE and the calculation can be started or stopped with *enable*. Depending on the parameterization of the inputs, a ramp function is calculated. Various options can be set by selecting the inputs. If *mode* is at False, the ramp is braked using a special entered deceleration time (*fastRampdownTime*) and without jerk limiting. This means that the ramp can be effectively decelerated for an Emergency Stop – this is also possible while accelerating. If *mode* is at True, then the drive is braked normally with the entered deceleration time (*rampdownTime*). The acceleration is always calculated with the entered acceleration time (*rampupTime*) independent of *mode.* In order to calculate the acceleration and deceleration stages with reference to the entered *nominalValue*, then the sampling time must be specified using the *taskMode* input. In order to reach the setpoint (*x*), the appropriate interval values are output (at *y*) and the actual acceleration is output (at *ya*). If input *enableRounding* is set to TRUE, the ramp is calculated with jerk limiting. The rounding-off is determined by the rounding-off time (*roundingTime*). The zero crossing (when the input *x* changes direction) can be parameterized with or without rounding. As soon as the output value is zero this is displayed using the *setpointZero* output. Further, when accelerating output *acceleration* is set, when decelerating, output *deceleration* is set and for a constant value *setpointReached* is set. If the block is in the enable mode, then output *busy* is set to TRUE. If the *hold* input is set to TRUE, the ramp-function generator is stopped (held) and the actual acceleration is ramped-down to zero. As long as *hold* is TRUE, setpoint x is ignored and output *ya* is set to zero.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Graphic representation of the function

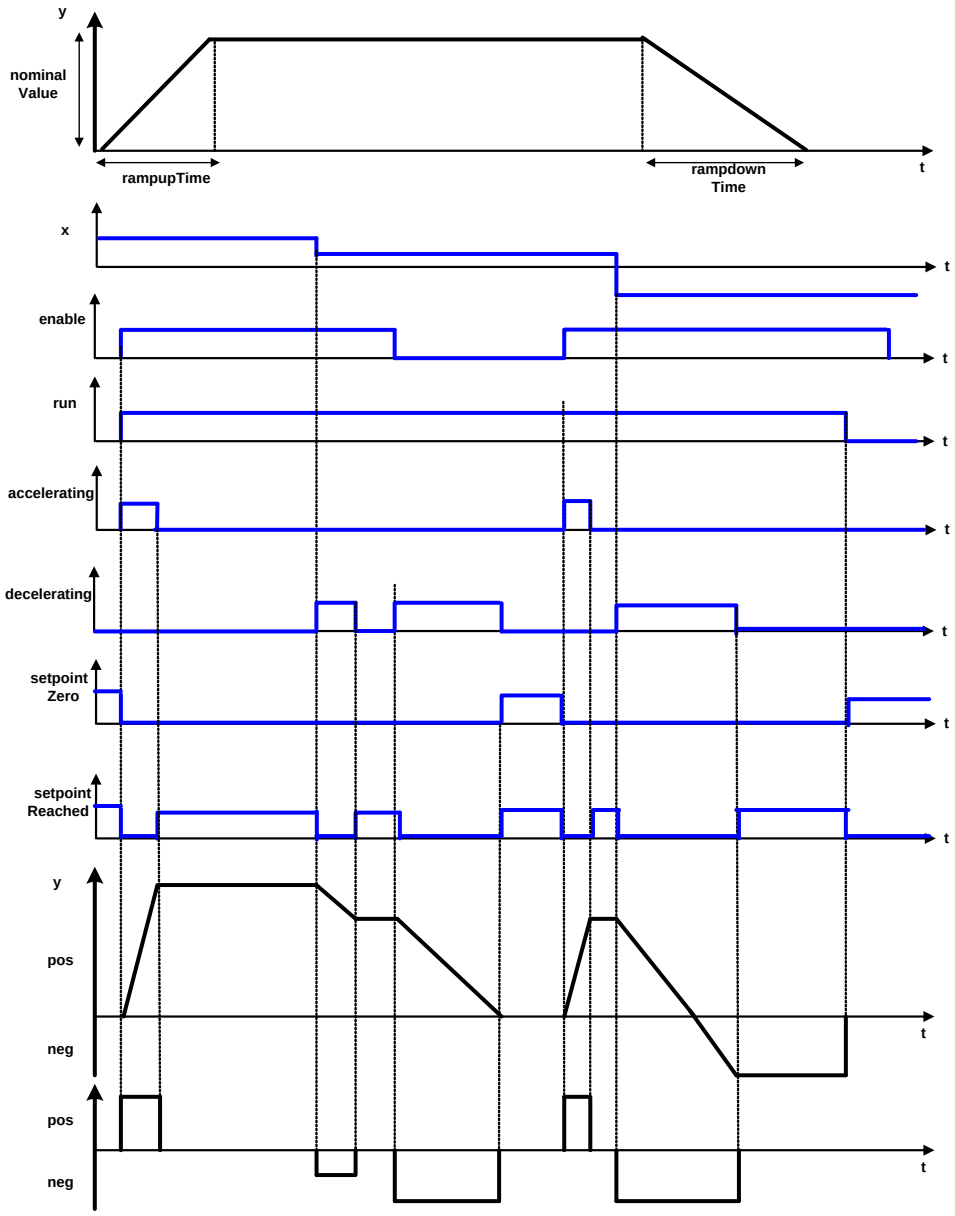
Figure 2-6: With rounding-off (enableRounding = TRUE)



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Figure 2-7: Without rounding-off (enableRounding = FALSE)



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Figure 2-8: With rounding-off and Emergency Stop function (mode = FALSE)

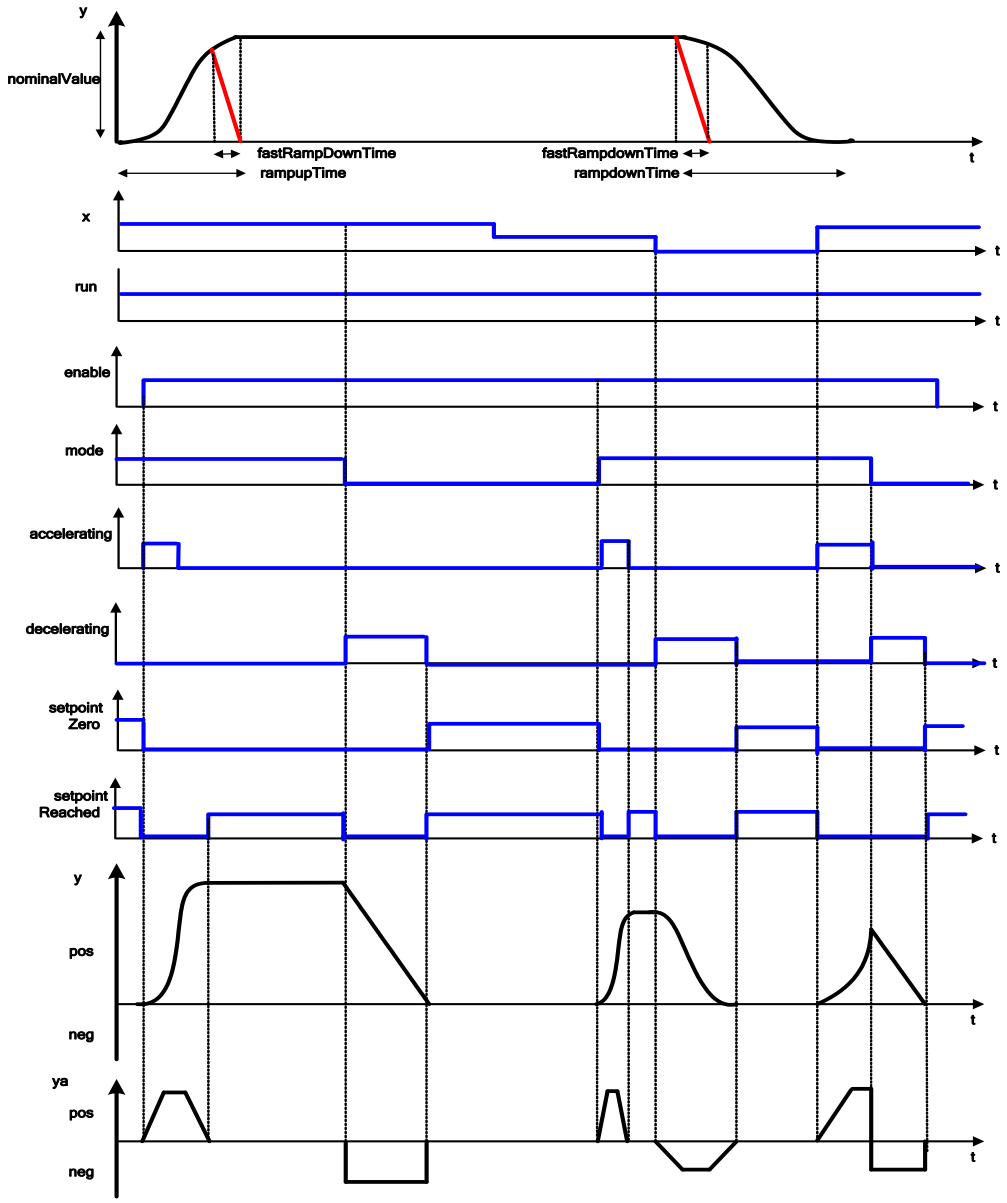
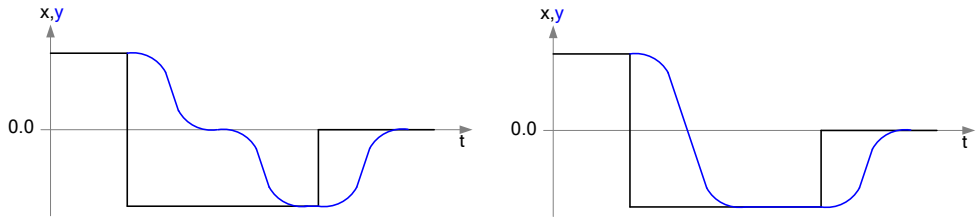


Figure 2-9: With or without zero rounding



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Errors

Error messages are indicated by the status of the outputs *error* and *errorID*: *error* = TRUE and *errorID* <> 16#0000_0000.Error = TRUE

ErrorID	Description
16#0000_0000	No error
16#010E_8001	The function block is called in an incorrect task.
16#010E_8005	Invalid input parameter *nominalValue = 0.0*

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

2.7 FCCalculateDistance

Calculates the braking distance and time, taking into consideration the dynamic parameters of the motion sequence.

Block name

FCCalculateDistance

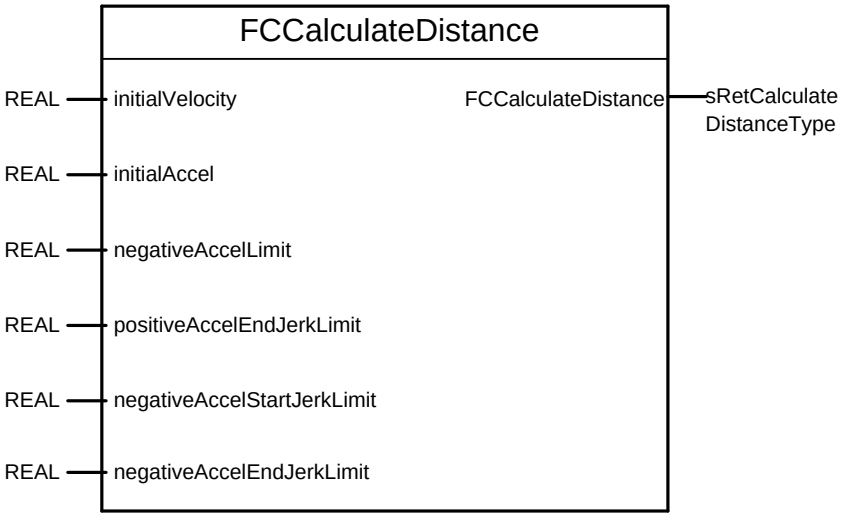
Task

This block calculates the braking distance and the associated time for the axis to brake down to standstill (zero speed) - taking into account the actual motion vector (velocity, acceleration, jerk).

Integration in the run-time system

The function can be called in any task.

Graphic representation of the block



Block parameters

Input parameters

Name	Data Type	Initial value	Description
initialVelocity	REAL	0.0	[LU / t] actual velocity of the axis – *initialVelocity* > 0.0
initialAccel	REAL	0.0	[LU/ t²] actual acceleration of the axis - if *initialVelocity* = 0.0 then *initialAccel* must be equal = 0.0.
negativeAccelLimit	REAL	0.0	[LU / t²] maximum deceleration – *negativeAccelLimit* < 0.0 and the absolute value must be greater than or equal to *initialAccel*.
positiveAccelEndJerkLimit	REAL	0.0	[LU / t³] jerk to decrease the acceleration – *positiveAccelEndJerkLimit* <= 0.0

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Name	Data Type	Initial value	Description
negativeAccelStartJerkLimit	REAL	0.0	[LU / t³] jerk to increase the acceleration – *positiveAccelEndJerkLimit* <= 0.0
negativeAccelEndJerkLimit	REAL	0.0	[LU / t³] jerk to reduce the deceleration – *negativeAccelEndJerkLimit* >= 0.0.

Output parameters

Name	Data Type	Initial-value	Description
FCCalculateDistance	sRet Calculate DistanceType		Return structure of the function with subsequently described elements
.boError	BOOL	FALSE	FALSE: No error TRUE: An error has occurred, refer to error identification
.b32ErrorID	DWORD	16#0000_0000	Error identification, refer to error messages b32errorID <= 16#000_8000: warning b32errorID >= 16#0000_8000: error
.r32GrossDistance	REAL	0.0	Calculated braking distance
.r32GrossTime	REAL	0.0	Calculated braking time

Data types and structures

Name of the structure	Contents
sRetCalculateDistanceType	This structure includes the output parameter of the FCCalculateDistance function. Declared in: LConLib.fConvTools

sRetCalculateDistance

The following type definition applies:

I/O	Element	Data type	Description
OUT	boError	BOOL	FALSE: No error TRUE: An error has occurred, refer to error identification
OUT	b32ErrorID	DWORD	Error identification, refer to error messages b32errorID <= 16#000_8000: warning b32errorID >= 16#0000_8000: error
OUT	r32GrossDistance	REAL	The gross distance needed for axis stop with given inputs.
OUT	r32GrossTime	REAL	The gross time need for axis stop.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Functionality

The motion profile is sub-divided into four segments for jerk-limited stopping of an axis. In this case, a certain initial velocity and initial acceleration are used as basis.

- In the first segment (S2) the initial acceleration (initialAcceleration) is decreased with the specified jerk (positiveAccelEndJerkLimit). The velocity increase is taken into account.
- In the second segment (S3) it is checked as to whether the maximum deceleration is reached with the specified jerk (negativeAccelStartJerkLimit). The maximum deceleration is calculated.
- In the third segment (SD), the axis is braked with maximum deceleration (negativeAccelLimit).
- In the fourth segment (S4) the actual acceleration is reduced with the specified jerk (negativeAccelEndJerkLimit).

Starting from the parameterization of the block, not all of the segments are always taken into account in the calculation. An overview of the use is given in the following table. If a return parameter is zero, then the corresponding segment is deactivated.

	da	dv	ds	dt
S2	neg	pos	pos	T2
S3	neg	neg	pos	T3
SD	-	neg	pos	TD
S4	pos	neg	pos	T4

Description of the elements in the table:

a – Acceleration v – Velocity

s – Distance t - time

d in the designations specifies that only changes to the values are taken into account – not the absolute results. The sign indicates an increase or decrease of the corresponding value.

Based on the input parameters, the segments are taken into consideration as follows:

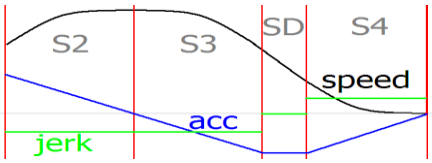
	initialVelocity						
	> 0			= 0			< 0
	initialAccel			*initialAccel*			*initialAccel*
	> 0	= 0	< 0	> 0	= 0	< 0	-
S2	1	0	0	1	0	error	error
S3	1	1	1*	1	0	error	error
SD	1*	1*	1*	0	0	error	error
S4	1	1	1	1	0	error	Error

**dependent on initialVelocity and negativeAccelLimit*

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Graphic representation of the function

The following diagram shows the motion phases relevant for the calculation starting from a positive initial velocity and acceleration.



In this example, all 4 segments are required for the calculation.

Errors

Error messages are indicated by the status of the outputs *error* and *errorID*: *error* = TRUE and *errorID* <> 16#0000_0000.

ErrorID	Description
16#0000_0000	No Error
16#010F_8005	Invalid input parameter The specification of the input parameter was not maintained as described. *negativeAccelLimit* >= 0.0 *positiveAccelEndJerkLimit* > 0.0 *negativeAccelStartJerkLimit* > 0.0 *negativeAccelEndJerkLimit* < 0.0 *initialVelocity* < 0.0 (*initialVelocity* = 0.0) and (*initialAccel* < 0.0) (ABS(*initialAccel*) > ABS(*negativeAccelLimit*)) and *initialAccel* < 0.0) If the input parameters lie in a valid value range, then it is not possible to stop the axis with limited jerk using the specified dynamic parameters.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

2.8 FCEvaluateModulo

Calculates the difference between two sampling values of a signal with overflow property.

Block name

FCEvaluateModulo

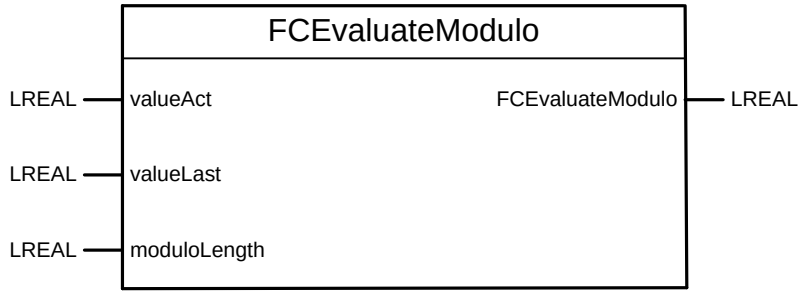
Task

The block calculates the difference of an input signal between two calls taking into account the overflow behavior, e.g. to determine the position difference of modulo axes.

Integration in the run-time system

The function can be called in any task.

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Block parameters

Input parameters

Name	Data type	Initial value	Description
valueAct	LREAL	0.0	[-] actual sampling value [n]
valueLast	LREAL	0.0	[-] previous sampling value [n – 1]
moduloLength	LREAL	0.0	[-] modulo length, corresponds to #POSAXIS. activeconfigdata.Modulo.Length. For moduloLength < 0.0, the calculation is made without modulo correction.

Output parameters

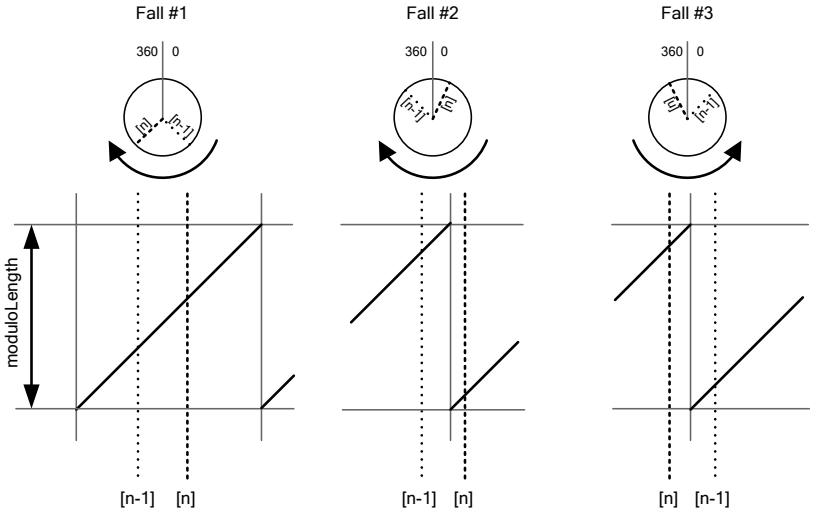
Name	Data type	Initial value	Description
FCEvaluateModulo	LREAL	-	[-] calculated difference

Functionality

The block calculates the position difference between two block calls and corrects the calculated value taking into account the modulo length. The evaluation is realized by comparing the actual and last position value referred to the modulo length.

In order that the block correctly operates, it must be ensured that a modulo length is run through in not less than 2 sampling cycles – as otherwise, reliable detection is not possible.

Fig. 2-10



Case #1:

Value [n] > Value [n – 1], no modulo overflow, no modulo correction.

FCEvaluateModulo = Value [n] – Value [n – 1]

Case #2:

Value [n] < Value [n - 1], modulo overflow in positive direction

FCEvaluateModulo = Value [n] – Value [n – 1] + moduloLength

Case #3:

Value [n] > Value [n – 1], modulo overflow in negative direction

FCEvaluateModulo = Value [n] – Value [n – 1] - moduloLength

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

In the examples, the modulo length is 360:

Table 2-3

valueLast	valueAct	Raw value	deltaPos	deltaNeg	Result
130	330	200	200	-160	-160
330	130	-200	160	-200	160
330	150	-180	180	-180	180
330	330	0	0	0	0
150	330	180	180	-180	180
330	170	-160	200	-160	-160
170	330	160	160	-200	160

Error messages

The block does not generate any error messages.

If the input parameters are incorrect, then the difference is calculated without taking the modulo into consideration.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

3 Drive functions

3.1 FBFrictionMeasurement – Friction characteristic

The friction characteristic is measured in SIMOTION

Block name

FBFrictionMeasurement

Task

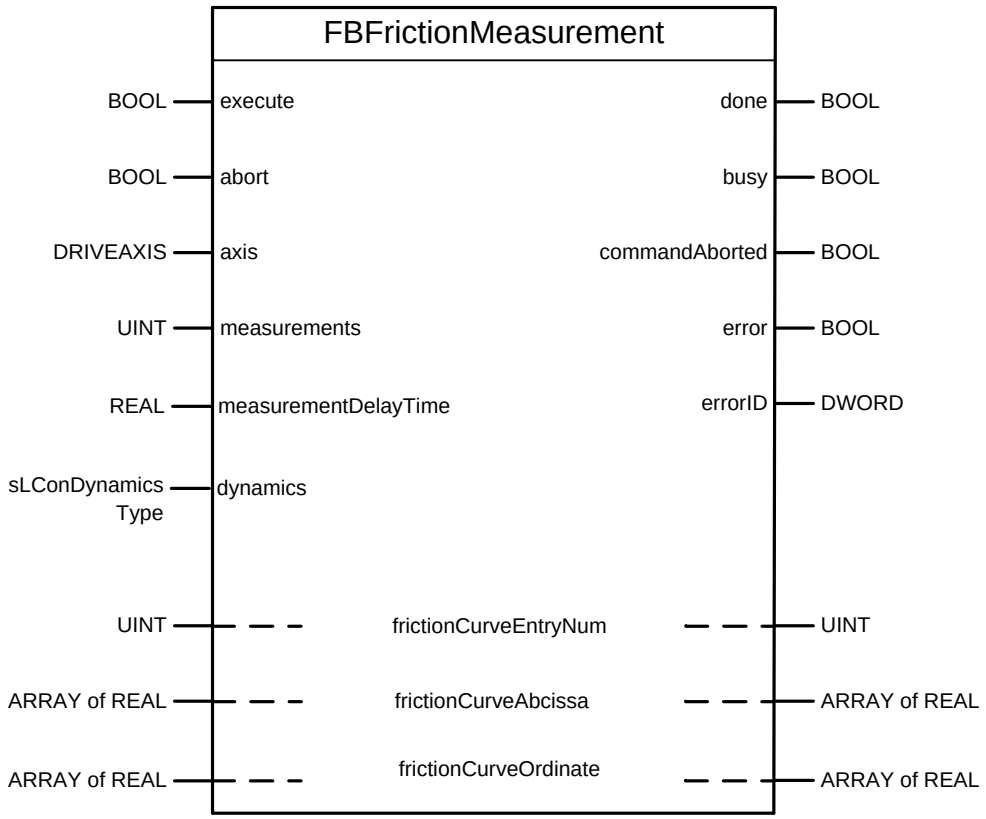
The function block runs up to the speeds, stored in the input array, and stores the corresponding average of the actual moments in the output array. E.g. the stored friction characteristic can be in the winder function block.

Integration in the run-time system

The function block must be called in a cyclic task (Background, Servo, IPO, IPO_2 or Timer).

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Graphic representation of the block



Block parameters

Input parameters

Name	Data type	Initial value	Description
execute	BOOL	FALSE	FALSE->TRUE starts the friction measurement
abort	BOOL	FALSE	FALSE->TRUE: Sequence interrupt and axis stopped
axis	DRIVEAXIS	TO#NIL	Axis for which the friction characteristic should be measured [min. speed axis]
measurements	UINT	50	Number of values to generate the average value [0 – 50]
measurementDelayTime	REAL	100	[ms] Delay time before starting the measurement at each speed level
dynamics	sLConDynamicsType	-	Axis dynmics from config structure

Input/Output parameters

Name	Data type	Initial value	Description
frictionCurveEntryNum	UINT	0	Number of grid points (Input): [2 ≤ frictionCurveEntryNum ≤ LCONLIB_MAX_IDX_A+1]
frictionCurveAbscissa	ARRAY [0.. LCONLIB_MAX_IDX_A] OF REAL	-	[rpTU ; °/TU] Abscissa values for the friction characteristic (speed; velocity) [maximum' LCONLIB_MAX_IDX_A+1' x-y points], corresponds to the configured units of the winder axis ([rev / TU] or [° / TU])
frictionCurveOrdinate	ARRAY [0.. LCONLIB_MAX_IDX_A] OF REAL	-	[M] Ordinate values for the friction characteristic (torque) [maximum ' LCONLIB_MAX_IDX_A+1' x-y points]

Output parameters

Name	Data type	Initial value	Description
done	BOOL	FALSE	FALSE: A friction characteristic is not generated TRUE: Measurement completed, friction characteristic generated
busy	BOOL	FALSE	FALSE: A measurement is not made TRUE: A measurement is made
CommandAborted	BOOL	FALSE	FALSE: Measurement was not interrupted TRUE: Measurement was prematurely interrupted
error	BOOL	FALSE	FALSE: No error TRUE: An error has occurred, refer to error identification
errorID	DWORD	16#0000_0000	Error identification, refer to error messages errorID <= 16#000_8000: warning errorID >= 16#0000_8000: error

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Functionality

Dependent on a request entered via the *request* input, the function block measures the friction characteristic of the axis *axis*. If the axis is inactive, then it is enabled and after the measurement has been completed it is inhibited again; for an active axis, this remains active. With the measurement, the velocities or speeds – that should be approached one after the other - are entered via the array *frictionCurveAbscissa*. At each velocity or speed level that is reached (constant move) and after the delay time *measurementDelayTime* the actual torque is saved in the array *frictionCurveOrdinate*. In order to have the actual torque in the TO axis, the technology data block must be activated. Parameter *frictionCurveEntryNum* specifies how many values are to be taken. The average value is calculated over several cycles. The number of measurements involved in generating the average value can be defined using the input *measurements*. In this case, the input is limited to 50 measurements. Output *busy* is set to True during the measurement. After all of the velocities or speeds have been approached, and the measurement completed, the *done* output is set to TRUE. If an error occurs during the measurement, this is displayed at the *error* output and the associated *errorID*. If, at the input, the abort signal *abort* is set to TRUE, the measurement is interrupted and the axis stopped. For errors involving the system commands (_enableAxis, _move, _stop, _disableAxis) and when the measurement is interrupted (aborted) by *abort*, the *commandAborted* output is set.

After the friction characteristic measurement has been successfully completed the values are in the array *frictionCurveAbscissa.*

Note

The units of the velocity or speed in the array *frictionCurveOrdinate*, should correspond to the units of the TO axis. If possible, [RPM] should also be plotted because FB Torque Pre Control calculates the friction torque with these units.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Errors

Error messages are indicated by the status of the outputs *error* and *errorID*: *error* = TRUE and *errorID* <> 16#0000_0000.

ErrorID	Description
16#0000_0000	No error
16#0108_C001	Non-compatible TO at the *axis* input
16#0108_C002	Technology data block is not activated
16#0108_8100	Error when enabling the axis
16#0108_8101	Error for a motion command
16#0108_8103	Error when stopping the axis
16#0108_8104	Error when inhibiting the axis
16#0108_C000	Internal program error

3.2 FBEnableInterface – Activate Interfaces

Function block for selective switching of the axis position, speed or torque interfaces.

Name of function block

FBEnableInterface

Task

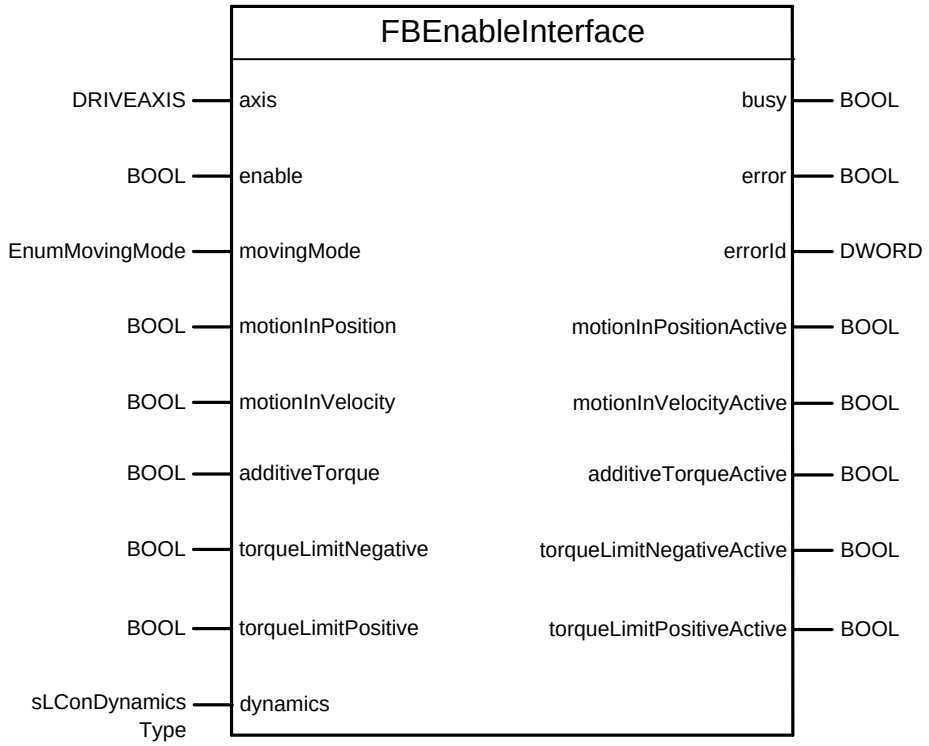
The function block activates or deactivates the cyclical interface for position, velocity and torque. The interfaces are necessary e.g. for controlling of an axis winder or a drive with overlaid tension control.

Integration into the runtime system

The function block must be called in a cyclic task (Background, Servo, IPO, IPO_2 or Timer).

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Graphic presentation of the block



Block parameters

Input parameters

Name	Data type	Initial value	Description
axis	DRIVEAXIS	-	Technology object axis
enable	BOOL	FALSE	TRUE: the function block is active, interfaces will be activated/deactivated accordingly TRUE>FALSE: all interfaces will be deactivated FALSE: function block inactive
movingMode	enumMoving Mode	SPEED_ CONTROLLED	Mode selection for axis control. Only relevant, if the axis is a positioning axis.
motionInPosition	BOOL	FALSE	TRUE: position interface is activated FALSE: position interface is deactivated The position interface is only on available for positioning axes. The velocity interface has priority over the position interface. No messages are output if the command is not allowed.
motionInVelocity	BOOL	FALSE	TRUE: velocity interface is activated FALSE: velocity interface is deactivated The velocity interface has priority over the position interface.
additiveTorque	BOOL	FALSE	TRUE: Interface for additive torque is activated FALSE: interface for additive torque is deactivated Technology data block is mandatory. No messages are output if the command is not allowed.
torqueLimitNegative	BOOL	FALSE	TRUE: interface for negative torque limit is activated FALSE: interface for negative torque limit is deactivated Technology data block is mandatory. No messages are output if the command is not allowed.
torqueLimitPositive	BOOL	FALSE	TRUE: interface for positive torque limit is activated FALSE: interface for positive torque limit is deactivated Technology data block is mandatory. No messages are output if the command is not allowed.
dynamics	sLConDynamic sType	-	Axis dynamics from config structure

Output parameters

Name	Data type	Initial value	Description
busy	BOOL	FALSE	FALSE: no processing active TRUE: at least one interface is active or being activated/deactivated or *enable = TRUE.*
error	BOOL	FALSE	FALSE: No error TRUE: An error has occurred, refer to error identification All interfaces will be deactivated.
errorID	DWORD	16#0000_0000	Error identification, refer to error messages errorID <= 16#000_8000: warning errorID >= 16#0000_8000: error

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Name	Data type	Initial value	Description
motionInPosition Active	BOOL	FALSE	TRUE: The interface is active. FALSE: The interface is inactive.
motionInVelocity Active	BOOL	FALSE	TRUE: The interface is active. FALSE: The interface is inactive.
additiveTorque Active	BOOL	FALSE	TRUE: The interface is active. FALSE: The interface is inactive.
torqueLimitNegative Active	BOOL	FALSE	TRUE: The interface is active. FALSE: The interface is inactive.
torqueLimitPositive Active	BOOL	FALSE	TRUE: The interface is active. FALSE: The interface is inactive.

Functionality

The Function Block activates and deactivates interfaces on a SIMOTION axis. The input parameter *axis* transfers the name of the axis on which the interfaces should be activated. Following input parameters de- or activate the interfaces:
motionInVelocity controls the MotionIn interface with velocity-Reference,
motionInPosition controls the MotionIn interface with position-reference,
additiveTorque controls the additive torque and
torqueLimitNegative/torqueLimitPositive are controlling the lower and upper torque limits.
The interfaces are getting activated/deactivated in the following sequence:

1.) Interface for velocity/ position (_runvelocitybasedmotionin / _runPositionBasedMotionIn / _move)

System variable: *axis*.defaultmotionin.velocity / *axis*.defaultmotionin.position

2.) Interface for additive torque (_enableaxisadditivetorque / _disableaxisadditivetorque)

System variable: *axis*.defaultadditivetorque

3.) Interface for negative torque limit (_enableaxistorquelimitnegative / _disableaxistorquelimitnegative)

System variable: *axis*.defaulttorquelimitnegative

4.) Interface for positive torque limit (_enableaxistorquelimitpositive / _disableaxistorquelimitpositive)

System variable: *axis*.defaulttorquelimitpositive

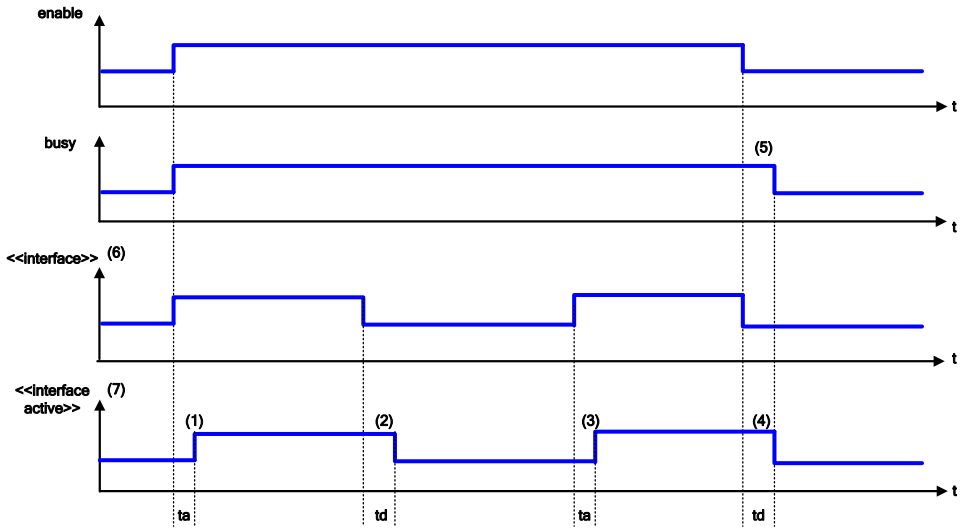
While calling the commands, the respective system variables are set to zero once. Exception is the activation of the MotionInPosition where the actual position is set. Is *enable* set to FALSE all interfaces become deactivated. As long as the function block is enabled the interfaces will always be activated and deactivated according to the respective inputs.

If an error occurs, the output *error* is set to TRUE and the identification is shown on output *errorID*.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Graphic representation of the function

Figure 3-1: Graphic representation of the function



1.) Interface will be activated, *ta* is the required activation time.

2.) Interface will be deactivated as the command is reset, *td* is the required deactivation time.

3.) Interface will be activated as the command is set once again, *ta* is the required activation time.

4.) Interface will be deactivated as the function block enable is switched off, *td* is the required deactivation time.

5.) The output *busy* is set to *FALSE* when **all** interfaces are inactive.

6.) And 7.) The sequence for activation/deactivation is the same for all (velocity, position, additive torque and torque limit) interfaces.

Note

For activation of the motion Interface, the current values of UserDefaultDynamics are taken and can not be changed later. For changing the dynamic values during run, without calling the command again it is recommended to raise the userDefaultDynamics and control the dynamic of the axis with plusLimitsOfDynamics / minusLimitsOfDynamics.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Errors

Error messages are indicated by the status of the outputs *error* and *errorID*:*error* = TRUE and *errorID* <> 16#0000_0000.

ErrorID	Description
16#0000_0000	No error
16#0109_810C	Error while activating the MotionIn interface with position-reference. The return value of the command _runPositionBasedMotionin is not 0.
16#0109_8105	Error while activating the MotionIn interface with velocity-reference. The return value of the command _runVelocityBasedMotionin is not 0.
16#0109_8106	Error while activating the additive torque interface. The return value of the command _disableAxisAdditiveTorque is not 0.
16#0109_8107	Error while activating the negative torque limit interface. The return value of the command _disableAxisTorqueLimitNegative is not 0.
16#0109_8108	Error while activating the positive torque limit interface. The return value of the command _enableAxisTorqueLimitPositive is not 0.
16#0109_8109	Error while deactivating the positive torque limit interface. The return value of the command _disableaxistorquelimitpositive is not 0.
16#0109_810A	Error while deactivating the negative torque limit interface. The return value of the command _disableaxistorquelimitnegative is not 0.
16#0109_810B	Error while deactivating the additive torque interface. The return value of the command _disableAxisadditiveTorque is not 0.
16#0109_8101	Error while deactivating the MotionIn interface. The return value of the command _move is not 0.
16#0109_C013	Time monitoring (watchdog) when switching the interfaces. An interface was not able to be activated or deactivated within one second.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

4 Closed-loop control functions

4.1 FBPT1 – PT1 element

PT1 filter that can be set.

Block name

FBPT1

Task

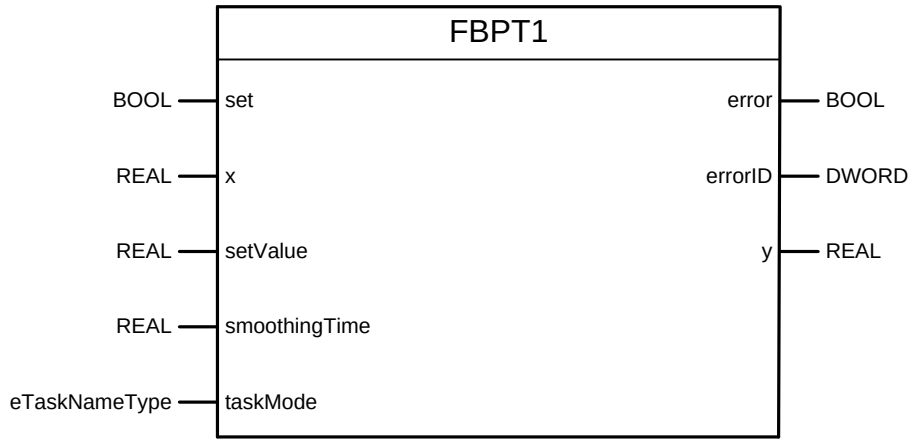
This function block has the function of a PT1 filter. The input value is output filtered. The filter effect can be influenced by the entered smoothing time. Further, it is possible to set the output to a specific value.

Integration in the run-time system

The function block must be called in a synchronous task (Servo, IPO, IPO_2 or Timer).

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Graphic representation of the block



Block parameters

Input parameters

Name	Data type	Initial value	Description
x	REAL	0.0	[-] Input value to be filtered
SmoothingTime	REAL	100.0	[ms] Smoothing time Internal limiting in the standard mode – at least one cycle time
setValue	REAL	0.0	[-] Value to be set for S = TRUE
set	BOOL	FALSE	FALSE: Normal operation of the filter TRUE: Sets the output to the value SV

Converting Library

Name	Data type	Initial value	Description
taskMode	eTaskName Type	IPO_ SYNCHRONOUS_ TASK	Specification of the task in which the block is called. SERVO_SYNCHRONOUS_TASK IPO_SYNCHRONOUS_TASK IPO_SYNCHRONOUS_TASK_2 TIMER_INTERRUPT_TASK

Output parameters

Name	Data type	Initial value	Description
error	BOOL	FALSE	FALSE: No error TRUE: An error has occurred, refer to error identification
errorID	DWORD	16#0000_0000	Error identification, refer to error messages errorID <= 16#000_8000: warning errorID >= 16#0000_8000: error
y	REAL	0.0	[-] Filtered output value

Functionality

A value *(x)* is entered into the function block. This value is filtered using the smoothing time *(smoothingTime)* and output at output *(y)*. To precisely calculate the internal steps, the sampling time in which the block is processed is internally determined. Further, it is possible to permanently set the output to the value of the input *(setValue)* by setting the input *(set)*.

Interconnection possibility:

In order to generate a HOLD, output *(y)* should be connected to the input *(setValue)* using a variable. This means that when setting *set* to TRUE, the output value is maintained until *set* is again set to FALSE. If the smoothing time *(smoothingTime)* is set to zero, the filter is de-activated and the output value is equal to the input value.

63% of the input value is reached at the output after the smoothing time *(smoothingTime)*.

95% of the input value is reached at the output after three times (300%) of the smoothing time *(smoothingTime)*.

The proportional gain – internally set – is 1 and cannot be varied.

For an appropriately high *smoothingTime* /TA (*smoothingTime* /TA > 10) the transfer function corresponds to the following characteristic

$$Y(t) = X * (1 - e^{-t/\text{smoothingTime}}) \quad with \;\; t = n * TA$$

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

The discrete values are calculated using the following algorithm:

$$Y_n = Y_{n-1} + \frac{TA}{\text{smoothingTime}} * (X_n - Y_{n-1})$$

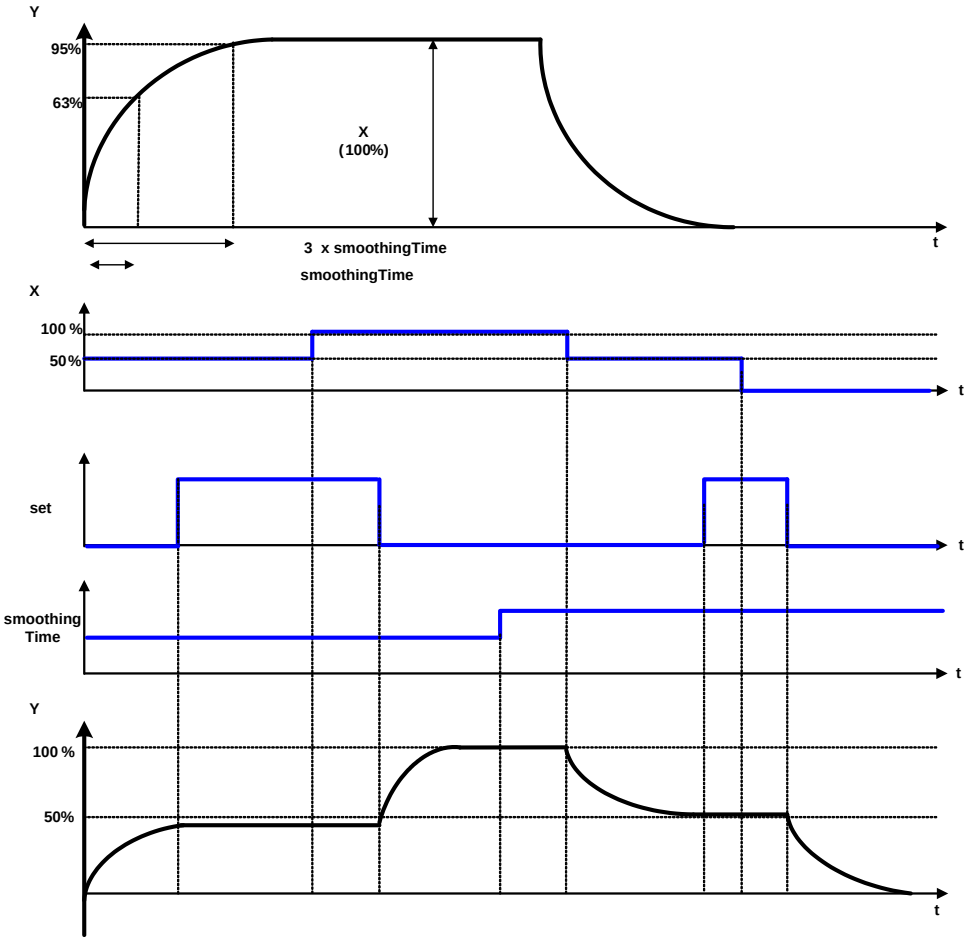
Y_n	Value of Y in sampling interval n
Y_{n-1}	Value of Y in sampling interval n – 1
X_n	Value of X in sampling interval n

Note

The higher the *smoothingTime* /TA, the lower the amplitude change at Y from one sampling instant to the next sampling instant. TA is the sampling time in which the function block is processed. *smoothingTime* is internally limited: T >= TA.

Graphic representation of the function

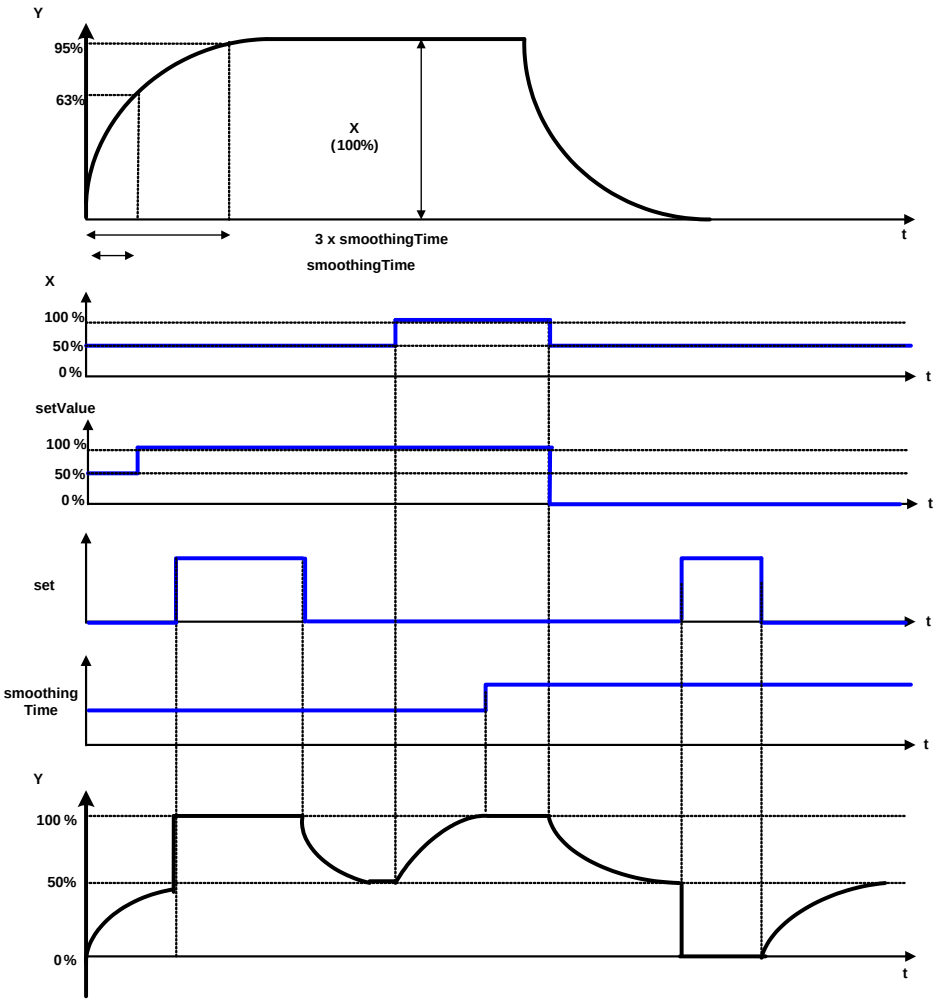
Figure 4-1: HOLD function (feedback from output y to setValue)



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Figure 4-2: Setting function (a fixed value is present at input setValue)



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Errors

Error messages are indicated by the status of the outputs *error* and *errorID*: *error* = TRUE and *errorID* <> 16#0000_0000.

ErrorID	Description
16#0000_0000	No error
16#0101_8001	The function block is called in an incorrect task.

4.2 FBINT – Integrator

Function block with integrating behavior, setting value function and output limiting

Block name

FBINT

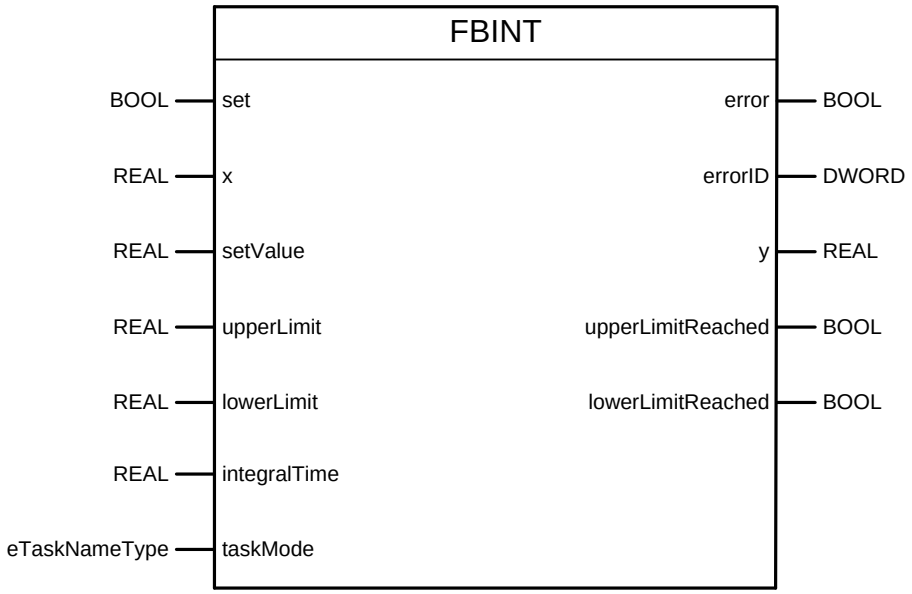
Task

The function block represents an integrator with upper and lower limiting. Further, the output can be set to a defined output value.

Integration in the run-time system

The function block must be called in a synchronous task (Servo, IPO, IPO_2 or Timer).

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Block parameters

Input parameters

Name	Data type	Initial value	Description
set	BOOL	FALSE	FALSE: Normal integrator function TRUE: Output is set value
x	REAL	0.0	[-] Input quantity, integration value
setValue	REAL	0.0	[-] Setpoint input for the setting function
upperLimit	REAL	1.0	[-] Upper integration limit
lowerLimit	REAL	0.0	[-] Lower integration limit
integralTime	REAL	5.0	[ms] Integration time TI [sampling time – 1000]
taskMode	eTaskName Type	IPO_ SYNCHRONOUS_ TASK	Specification of the task in which the block is called. SERVO_SYNCHRONOUS_TASK IPO_SYNCHRONOUS_TASK IPO_SYNCHRONOUS_TASK_2 TIMER_INTERRUPT_TASK

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Output parameters

Name	Data type	Initial value	Description
error	BOOL	FALSE	FALSE: No error TRUE: An error has occurred, refer to error identification
errorID	DWORD	16#0000_0000	Error identification, refer to error messages errorID <= 16#000_8000: warning errorID >= 16#0000_8000: error
y	REAL	0.0	[-] Output quantity, integration value
upperLimit Reached	BOOL	FALSE	FALSE: Upper limit not reached TRUE: Upper limit reached
lowerLimit Reached	BOOL	FALSE	FALSE: Lower limit not reached TRUE: Lower limit reached

Functionality

The function block integrates input quantity *x* with integration time *integralTime*. The result is output at *y*. The integrator is limited by the upper limit *upperLimit* and the lower limit *lowerLimit*. If the integrator reaches one of the limits, then this is connected to the output and saved as internal buffer value. Further, when the particular limit at the output is reached, this is indicated using *upperLimitReached* and *lowerLimitReached*. Using the setting function, which is activated with *set =True*, the output can be set to a pre-defined value *(setValue)*. If *set* is reset, then starting from the setting value, the integrator continues to integrate in the direction specified using input value *x*.

integralTime is internally limited with

$$TA \leq \text{integralTime} \leq 1000$$

Converting Library

The discrete values (TA is the sampling time in which the function block is processed) are calculated using the following algorithm:

$$Y_n = Y_{n-1} + \frac{TA}{\text{integralTime}} * X_n$$

Yn	Value of Y in sampling interval n
Yn-1	Value of Y in sampling interval n-1
X	Value of X in sampling interval n

Graphic representation of the function

Figure 4-3: Block diagram

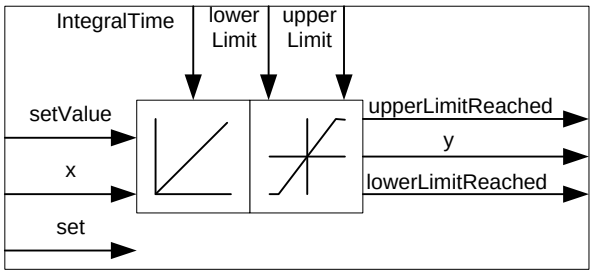


Table 4-1 Truth table for LL<LU

Set	Condition	y	upper Limit Reached	lower Limit Reached	Mode
0	lowerLimit < y_{n-1} + x * TA/IntegralTime < upperLimit	y_n	0	0	Integrating
0	y_{n-1} + x * TA/integralTime >= upperLimit	upper Limit	1	0	Integrator at the upper limit
0	y_{n-1} + x * TA/integralTime <= lowerLimit	lower Limit	0	1	Integrator at the lower limit
1	lowerLimit < setValue < upperLimit	set $Value_n$	0	0	Setting
1	setValue >= upperLimit	upper Limit	1	0	Integrator at the upper limit
1	setValue <= lowerLimit	lower Limit	0	1	Integrator at the lower limit

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Table 4-2 Truth table for lowerLimit > upperLimit

Set	Condition	Y	upper Limit Reached	lower Limit Reached	Mode
*	lowerLimit < y_{n-1}+x * TA/integralTime < upperLimit	y_n	0	0	Integrator limited by upper Limit

* = any

Errors

Error messages are indicated by the status of the outputs *error* and *errorID*: *error* = TRUE and *errorID* <> 16#0000_0000.

ErrorID	Description
16#0000_0000	No error
16#0102_8001	The function block is called in an incorrect task.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

4.3 FBDIF – Differentiating element

Differentiating element

Block name

FBDIF

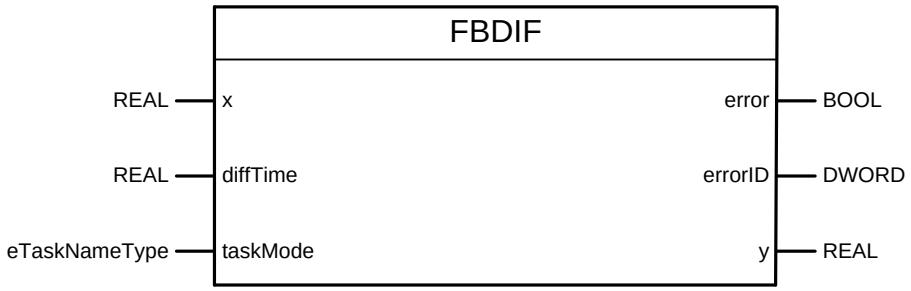
Task

Function block with differentiating behavior

Integration in the run-time system

The function block must be called in a synchronous task (Servo, IPO, IPO_2 or Timer).

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Block parameters

Input parameters

Name	Data type	Initial value	Description
x	REAL	0.0	[-] Input quantity
diffTime	REAL	0.0	[ms] Differentiating time constant
taskMode	eTaskName Type	IPO_ SYNCHRONOUS_ TASK	Specification of the task in which the block is called. SERVO_SYNCHRONOUS_TASK IPO_SYNCHRONOUS_TASK IPO_SYNCHRONOUS_TASK_2 TIMER_INTERRUPT_TASK

Output parameters

Name	Data type	Initial value	Description
error	BOOL	FALSE	FALSE: No error TRUE: An error has occurred, refer to error identification
errorID	DWORD	16#0000_0000	Error identification, refer to error messages errorID <= 16#000_8000: warning errorID >= 16#0000_8000: error
y	REAL	0.0	[-] Output quantity

Converting Library

Functionality

Output quantity *y* changes as a function of the rate of change of input quantity *x*, multiplied by the differential time constant *diffTime*. Discrete values are calculated using the following algorithm:

$$Y_n = (X_n - X_{n-1}) * \frac{diffTime}{TA}$$

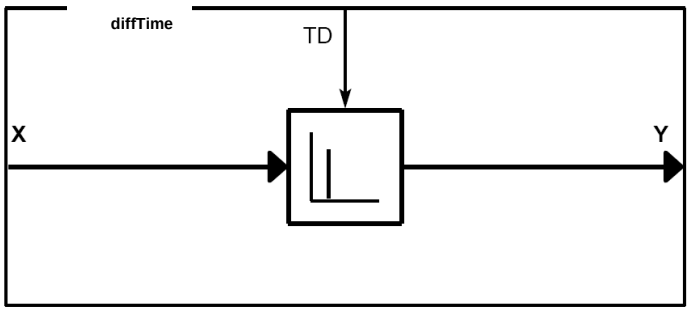
Yn	Value of Y in sampling interval Y
Xn	Value of X in sampling interval n
Xn-1	Value of X in sampling interval n-1

Note The higher the diffTime / TA the lower the amplitude change at y from one sampling instant to the next sampling instant. TA is the sampling time in which the function block is processed. diffTime is internally limited diffTime >= 0.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Graphic representation of the function

Figure 4-4: Block diagram



Errors

Error messages are indicated by the status of the outputs *error* and *errorID*: *error* = TRUE and *errorID* <> 16#0000_0000.

ErrorID	Description
16#0000_0000	No error
16#0103_8001	The function block is called in an incorrect task.

4.4 FBDT1 – DT1 filter

Function block with DT1 behavior

Block name

FBDT1

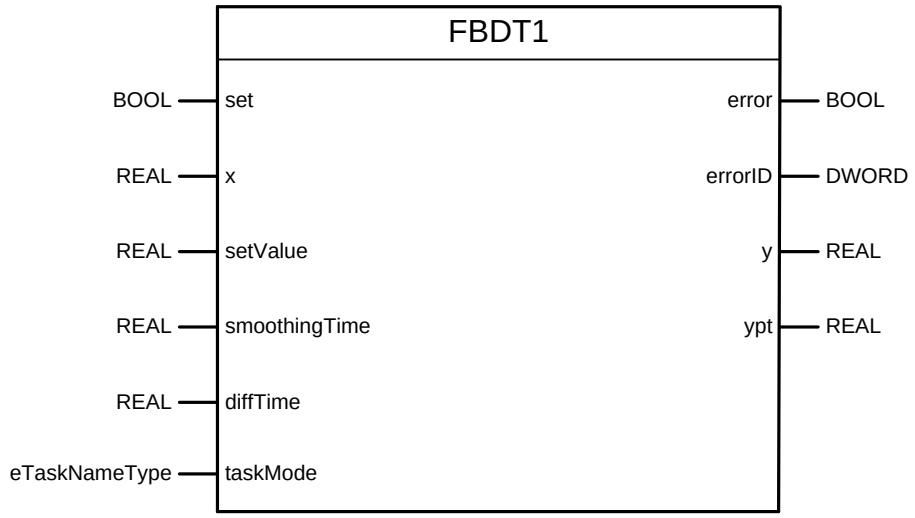
Task

Function block with differentiating behavior and smoothing. The DT1 output can be set.

Integration in the run-time system

The function block must be called in a synchronous task (Servo, IPO, IPO_2 or Timer).

Graphic representation of the block



Block parameters

Input parameters

Name	Data type	Initial value	Description
set	BOOL	FALSE	FALSE: Setting value has no influence on the output quantity of the DT1 element TRUE: Output value Y is set to the value of input SV and the output of the DT1 element corresponds to the equation below
x	REAL	0.0	[-] Input quantity
setValue	REAL	0.0	[-] Setting value
SmoothingTime	REAL	0.0	[ms] Smoothing time constant is internally limited :[SamplingTime – T1Max]
diffTime	REAL	0.0	[ms] Differentiating time constant is internally limited : [0.0 – TDMax]

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Name	Data type	Initial value	Description
taskMode	eTaskName Type	IPO_ SYNCHRONOUS_ TASK	Specification of the task in which the block is called. SERVO_SYNCHRONOUS_TASK IPO_SYNCHRONOUS_TASK IPO_SYNCHRONOUS_TASK_2 TIMER_INTERRUPT_TASK

Output parameters

Name	Data type	Initial value	Description
error	BOOL	FALSE	FALSE: No error TRUE: An error has occurred, refer to error identification
errorID	DWORD	16#0000_0000	Error identification, refer to error messages errorID <= 16#000_8000: warning errorID >= 16#0000_8000: error
y	REAL	0.0	[-] Output quantity DT1
ypt	REAL	0.0	[-] Output quantity smoothing element

Functionality

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Setting function not active (set = FALSE)

Input quantity x is dynamically delayed and the smoothing time constant *smoothingTime* is entered at a differentiating element and at block output YPT. The output quantity Y of the complete DT1 element is multiplied, with the same relationship of the rated of change of YPT (difference quotient) with differential time constant *diffTime*.

SmoothingTime defines the gradient of the down ramp of the output quantity. This specifies the time after which the transfer function – after smoothing and differentiation – has dropped to 37% of X-*diffTime*/ *smoothingTime*. For a sufficiently high *smoothingTime* /TA (*smoothingTime* /TA > 10), the transfer function corresponds to a characteristic of:

$$Y(t) = X * (diffTime / smoothingTime) * e^{-t/smoothingTime} \qquad mit\ t = n * TA$$

The discrete values are calculated using the algorithm:

$$Y1_n = \frac{diffTime}{smoothingTime} * (X_n - YPT_{n-1})$$

$$YPT_n = YPT_{n-1} + \frac{TA}{smoothingTime} * (X_n - YPT_{n-1})$$

YPTn	Value of YPT in sampling interval n
Yn	Value of Y in sampling interval n
Xn	Value of X in sampling interval n
YPTn-1	Value of Y2 in sampling interval n-1

The higher the *smoothingTime* /TA, the lower the amplitude change at Y and at YPT from one sampling instant to the next sampling instant. TA is the sampling time in which the function block is processed.

The higher the *diffTime* /TA, the higher the amplitude change at Y from one sampling instant to the next sampling instant. *DiffTime* and *smoothingTime* are internally limited:

0 <= *diffTime* <= MaxTD, TA <= *smoothingTime* <= MaxT1

Setting function active (set = TRUE)

When the setting function is active, setting value *setValue* is transferred to DT1 output Y; the output of the smoothing element is obtained as follows:

$$YPT_n = X_n - \frac{smoothingTime}{diffTime} * setValue_n \qquad for\ diffTime \neq 0$$

In this case, the internal limits for *smoothingTime*1 and *diffTime* apply. For *diffTime* = 0, the output quantities remain unchanged as long as *set* = TRUE.

Caution: Overcontrol (limits reached) can occur when the setting function is both active as well as also inactive!

Graphic representation of the function

Figure 4-5: Block diagram

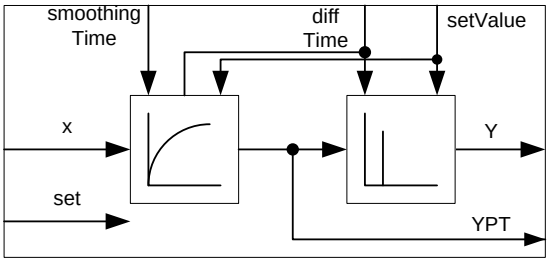
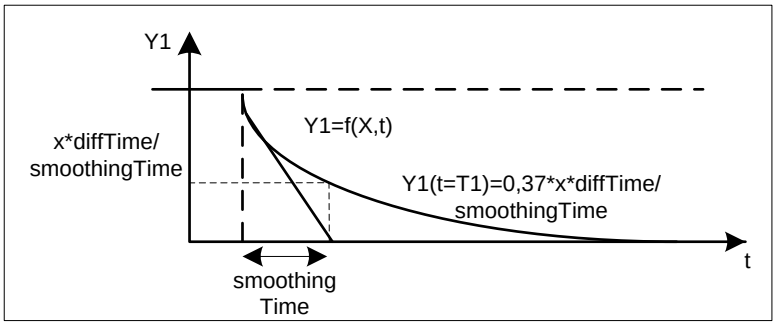


Figure 4-6: Transfer function



Errors

Error messages are indicated by the status of the outputs *error* and *errorID*: *error* = TRUE and *errorID* <> 16#0000_0000.

ErrorID	Description
16#0000_0000	No error
16#0104_8001	The function block is called in an incorrect task.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

4.5 FBLIM – Limiter with feedback signal

Limiter with settable limits

Block name

FBLIM

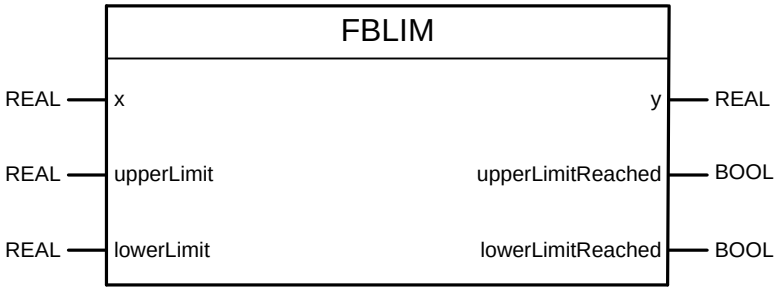
Task

Function block for limiting with settable upper and lower limits. If a limit is reached, this is displayed.

Integration in the run-time system

The function block can be called in any task.

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Block parameters

Input parameters

Name	Data type	Initial value	Description
x	REAL	0.0	[-] Input quantity to be limited
upperLimit	REAL	0.0	[-] Upper limit value
lowerLimit	REAL	0.0	[-] Lower limit value

Output parameters

Name	Data type	Initial value	Description
y	REAL	0.0	[-] Limited output quantity [*lowerLimit – upperLimit*]
upperLimitReached	BOOL	TRUE	FALSE: Upper limit not reached TRUE: Upper limit reached
lowerLimitReached	BOOL	TRUE	FALSE: Lower limit not reached TRUE: Lower limit reached

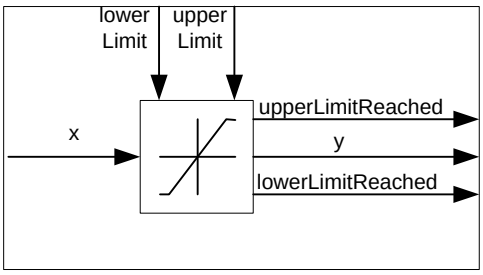
Converting Library

Functionality

The function block transfers input quantity *x* to its output *y*. The input quantity is limited dependent on *upperLimit* and *lowerLimit*.
If the input quantity reaches upper limit *upperLimit*, output *upperLimitReached* is set to TRUE.
If the input quantity reaches lower limit value *lowerLimit*, output *lowerLimitReached* is set to TRUE.
If the lower limit value is greater or equal to the upper limit value, output *y* is set to the upper limit value *upperLimit.*

Graphic representation

Figure 4-7



Error messages

If *upperLimitReached* and *lowerLimitReached* are both at TRUE, the lower limit *lowerLimit* is greater than upper limit *upperLimit*.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

4.6 FBLevelControl – Limit value monitor

Block name

FBLevelControl

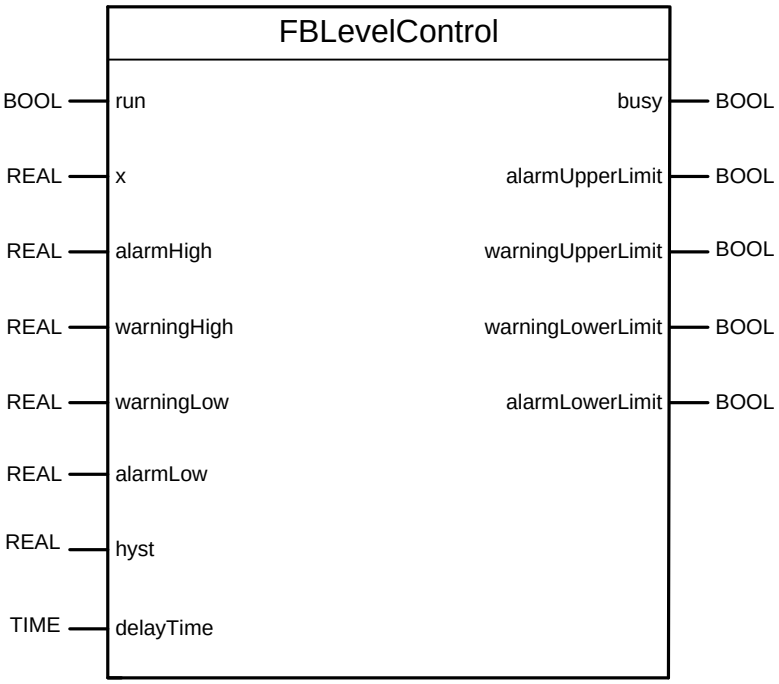
Task

This function block monitors an input value to identify whether it violates (exceeds, falls below) the parameterized limit values. The monitoring is only activated as long as the block is active.

Integration in the run-time system

The function block must be called in a cyclic task (Background, Servo, IPO, IPO_2 or Timer).

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Block parameters

Input parameters

Name	Data type	Initial value	Description
run	BOOL	FALSE	FALSE: Function is de-activated TRUE: Function is activated
x	REAL	0.0	[-] Input value to be monitored
alarmHigh	REAL	100.0	[-] Upper limit value, alarm
warningHigh	REAL	80.0	[-] Upper limit value, warning
warningLow	REAL	-80.0	[-] Lower limit value, warning
alarmLow	REAL	-100.0	[-] Lower limit value, alarm
hyst	REAL	0.0	[-] Hysteresis This value is always taken into account as absolute value.
delayTime	TIME	T#3s	[s] Delay Time to set outputs

Output parameters

Name	Data type	Initial value	Description
busy	BOOL	FALSE	FALSE: Function is de-activated TRUE: Function is activated
alarmUpperLimit	BOOL	FALSE	Alarm, upper limit value exceeded rX > rAH, the message is reset with rx < rAH – rHy.
warningUpper Limit	BOOL	FALSE	Warning, upper limit value exceeded rX > rWH, the message is reset with rx < rWH – rHy.
warningLower Limit	BOOL	FALSE	Warning, lower limit value exceeded rX < rWL, the message is reset with rx > rWL + rHy
alarmLowerLimit	BOOL	FALSE	Alarm, lower limit value exceeded rX < rAL, the message is reset with rx > rAL + rHy

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Functionality

This function block monitors an input value *x* to identify whether it violates (exceeds, falls below) parameterized limit values as long as it is active with *run* = TRUE. If the block is functioning, then this can be identified using the *busy* output. When the *warningHigh* value is exceeded, output *warningUpperLimit* is set after *delayTime*. The same applies when the *alarmHigh* value is exceeded with the associated output *alarmUpperLimit*. If the input value drops below the limit value *warningHigh* or *alarmHigh* minus the hysteresis value (*hyst)*, the particular output is again reset. The same applies for the lower limit monitoring with the limits *warningLow* and *alarmLow* as well as the associated outputs *warningLowerLimit* and *alarmLowerLimit*. However, in this case, the hysteresis (*hyst*) is added to the limit value. Warnings are not reset as a result of alarms.

4.7 FCDeadBand – Dead band element

Settable dead band that sets the value range – symmetrical around the zero point – to zero.

Block name

FCDeadBand

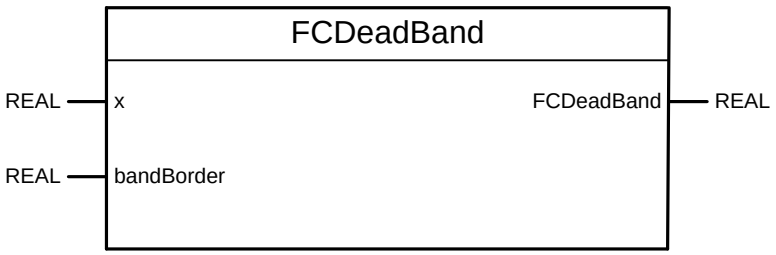
Task

The function generates a dead band whose values cannot be reached. All values in this value range –symmetrical around the zero point – are set to zero.

Integration in the run-time system

The function can be called in any task.

Graphic representation of the block



Block parameters

Input parameters

Name	Data type	Initial value	Description
x	REAL	0.0	[-] Input value that should be checked
bandBorder	REAL	0.0	[-] Limit value for dead band

Output parameters

Name	Data type	Initial value	Description
FCDeadBand	REAL	-	[-] Checked output value

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Functionality

If the input value (*x*) is located in the dead band range (-*bandBorder* < x < +*bandBorder*), the output value is set to zero. However, if it is at or below (*–bandBorder)* at output *input – bandBorder* is output. If the input value is at or above (*+bandBorder), x + bandBorder* is set to as output value. The dead band – symmetrical around the zero point – can be set using response value B.

Algorithm:

X + B for X <= -B
Y = 0 for -B < X < +B
X - B for X >= +B

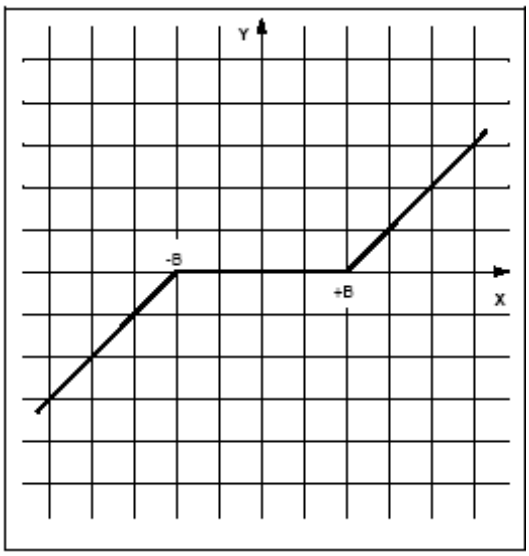
with the secondary condition B >= 0. For B < 0, all IN: OUT=IN.

X	Input quantity
B	Response threshold for the dead band
Y	Output quantity

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Graphic representation of the function

Figure 4-8: Transfer function



4.8 FCDeadZone – Dead zone element

Settable dead zone that sets the value range – symmetrical around the zero point – to zero.

Block name

FCDeadZone

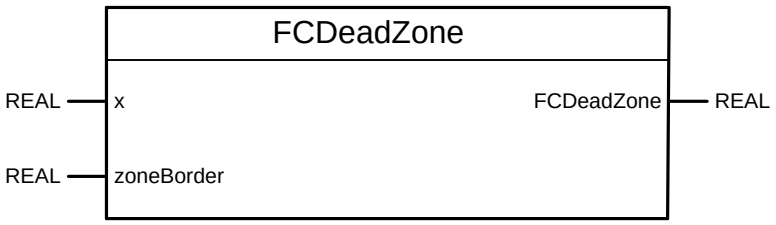
Task

The function generates a dead zone whose values cannot be reached. All values in the value range – symmetrical around the zero point – are set to zero.

Integration in the run-time system

The function can be called in any task.

Graphic representation of the block



Block parameters

Input parameters

Name	Data type	Initial value	Description
x	REAL	0.0	[-] Input value which should be checked
zoneBorder	REAL	0.0	[-] Limit value for dead zone

Output parameters

Name	Data type	Initial value	Description
FCDeadZone	REAL	-	[-] Checked output value

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Functionality

If the input value (*x*) lies in the range of the dead band (*-zoneBorder* < x < + *zoneBorder*), the output value (FCDeadZone) is set to zero. If the input value lies at / below (*–zoneBorder)* or at / above (+*zoneBorder)*, input x is output as output value. The dead zone – symmetrical around the zero point – can be set using the response value *zoneBorder*.

Algorithm:

X	for	X <= -TH
Y = 0	for	-TH < X < +TH
X	for	X >= +TH

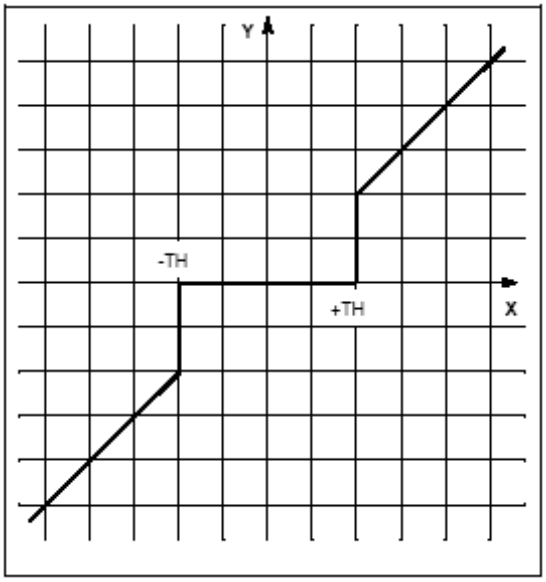
with the secondary condition TH >= 0. For TH < 0 for all the following applies X: Y = X.

X	Input quantity
TH	Response threshold for the dead zone
Y	Output quantity

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Graphic representation of the function

Figure 4-9: Transfer function



4.9 FBAverageValue – Average value generator

Floating average value generator

Block name

FBAverageValue

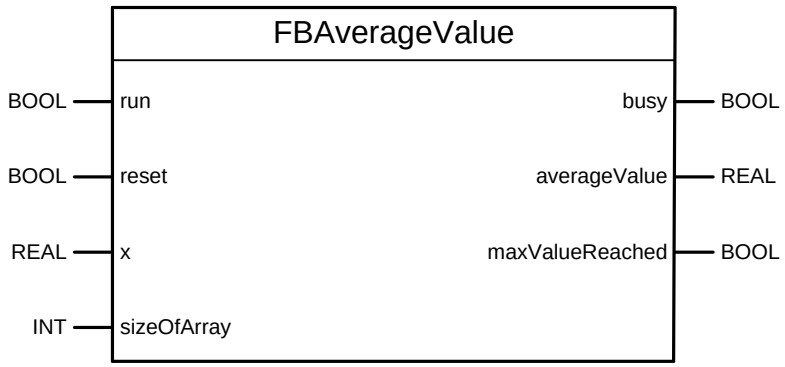
Task

The function calculates the floating average value for the values entered at the input.

Integration in the run-time system

The function block must be called in a cyclic task (Background, Servo, IPO, IPO_2 or Timer

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Block parameters

Input parameters

Name	Data type	Initial value	Description
run	BOOL	FALSE	FALSE: Function block is not active TRUE: Function block is active and calculates an average value
reset	BOOL	FALSE	TRUE: Resets the calculation
x	REAL	0.0	[-] Actual input value
sizeOfArray	INT	0	[-] Size of the field to save the value [1 – 50]

Output parameters

Name	Data type	Initial value	Description
busy	BOOL	FALSE	FALSE: FB does not calculate any values TRUE: FB calculates values
averageValue	REAL	0.0	[-] Calculated average value
maxValueReached	BOOL	FALSE	TRUE: maximum Values in the array *sizeOfArray* – Values are in the array

Functionality

The function block calculates the average value (*averageValue*) of several values that are entered and saved. Each time the block is called, the value (*x*) currently present is saved in a field (*sizeOfArray*). The larger the field, the more the average value moves. If the field size is very small, changes and fluctuations can be immediately identified in the average value, which is present at the output (*averageValue*). The function block is activated via the input *run* and calculates values from that instant. Similar to that for *reset*, the field size is accepted and the associated field counter is reset. The Output *maxValueReached* is set if sizeOfArray Values are in the array. With *reset* = TRUE the output *averageValue* is set to the input value *x.*

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

4.10 FBPIDController

PID controller with actual value smoothing and optional D component in the actual value branch.

Block name

FBPidController

Task

The function block has the following functionality:

- PID controller
- Actual value smoothing with PT1 element
- Scaling of the actual value signal
- Proportional, integral and derivative action component
- Optional derivative action component in the actual value arm
- “Anti-windup” integrator: If the controller output is limited, then the integrator is frozen when the limit is reached
- Pre-control signal
- Deadband filter for the system deviation
- Output scaling
- Output limiting

The block can be used as higher-level controller, e.g. for closed-loop tension control or closed-loop dancer roll control.

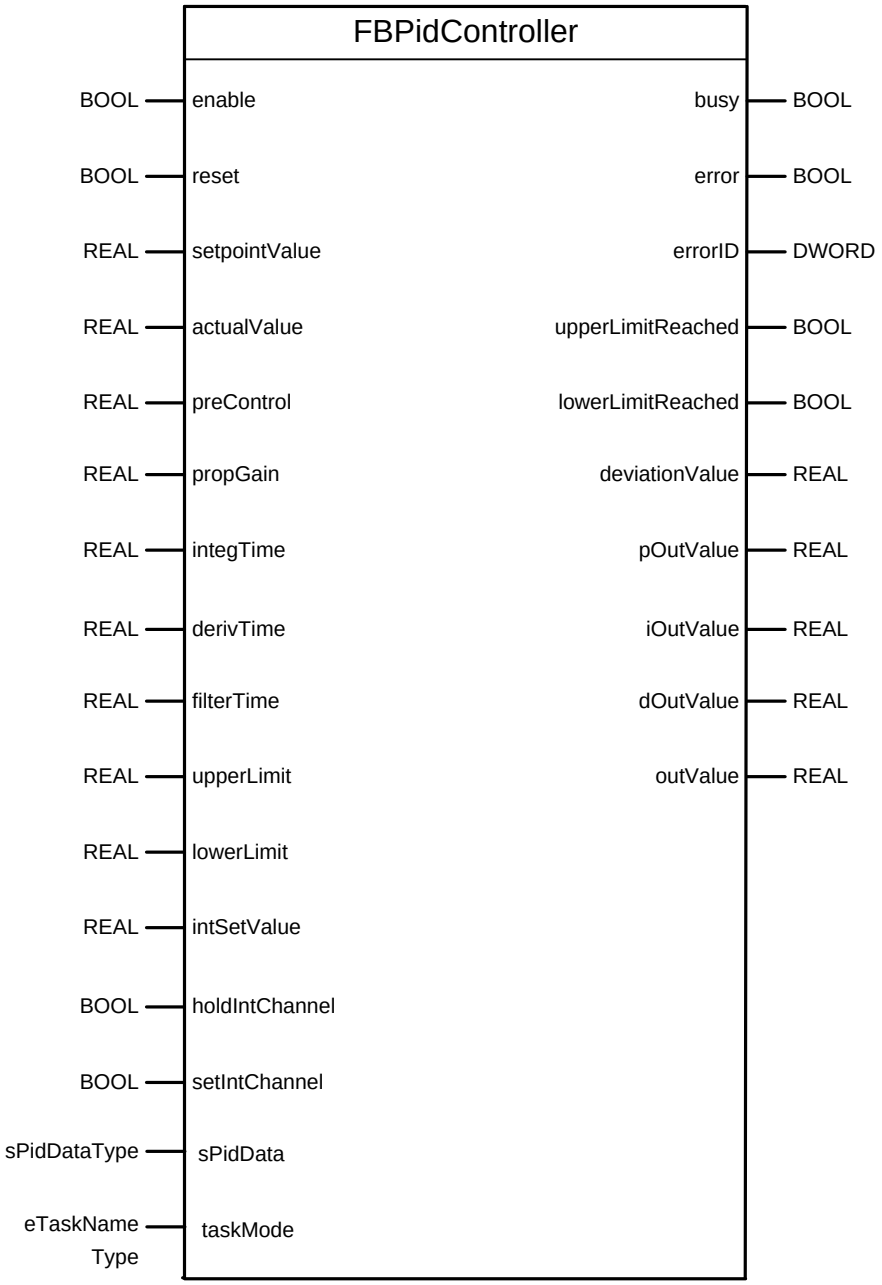
Integration in the run-time system

The function block must be called in a synchronous task (Servo, IPO, IPO_2 or Timer).

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Block parameters

Input parameter

Name	Data type	Initial value	Description
enable	BOOL	FALSE	TRUE: Controller, active, outputs are calculated FALSE: Controller inactive, all outputs are set to the initial values
reset	BOOL	FALSE	TRUE: The outputs are set to the initial values and the integrator to the starting value.
setpointValue	REAL	0.0	[-] Setpoint signal
actualValue	REAL	0.0	[-] Actual value signal
preControl	REAL	0.0	[-] Pre-control signal
propGain	REAL	2.0	[-] Controller gain
integTime	REAL	999.0	[ms] Time constant of the I component – is internally limited to cycle time / 2 as the lower limit value.
derivTime	REAL	4.0	[ms] Time constant of the D component – the value is internally limited to cycle time as lower limit value.
filterTime	REAL	50.0	[ms] Time constant of the actual value smoothing – this is internally limited to the cycle time as lower limit value. The smoothing is deactivated if values <= 0.0 are parameterized.
upperLimit	REAL	100.0	[-] Upper limit value of the control output. This must be greater than *lowerLimit*
lowerLimit	REAL	0.0	[-] Lower limit value of the control output.
intSetValue	REAL	0.0	[-] Integrator setting value
holdIntChannel	BOOL	FALSE	TRUE: I component is frozen
setIntChannel	BOOL	FALSE	TRUE: I component is set to the initial value
sPidDate	sPidDataType		Refer to the data type description sPidDataType
taskMode	eTaskName Type	IPO_ SYNCHRONOUS_ TASK	Execution level, in which the block is called. SERVO_SYNCHRONOUS_TASK IPO_SYNCHRONOUS_TASK IPO_SYNCHRONOUS_TASK_2 TIMER_INTERRUPT_TASK

Output parameter

Name	Data type	Initial value	Description
error	BOOL	FALSE	FALSE: No error TRUE: An error has occurred, refer to error identification All outputs have the initial values, errors can be reset using the reset function or rising edge at the enable input.
errorID	DWORD	16#0000_0000	Error identification, refer to error messages errorID <= 16#000_8000: warning errorID >= 16#0000_8000: error
busy	BOOL	FALSE	TRUE: Block is active
upperLimitReached	BOOL	FALSE	TRUE: Controller output has reached the upper limit, the integrator is limited
lowerLimitReached	BOOL	FALSE	TRUE: Controller output has reached the lower limit, the integrator is held
deviationValue	REAL	0.0	[-] System deviation after the deadband filter

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Name	Data type	Initial value	Description
pOutValue	REAL	0.0	[-] P component output, unlimited
iOutValue	REAL	0.0	[-] I component output, unlimited
dOutValue	REAL	0.0	[-] D component output, unlimited (if not in the actual value branch)
outValue	REAL	0.0	[-] Controller output, limited and scaled

Data types and structures

Name of the structure	Contents
sPidDataType	Configuration data of the PID controller Declared in: LConLib.fConvTools

sPidDataType

The following type declaration applies:

I/O	Element	Data type	Description
IN	boEnPChannel	BOOL	TRUE: P component is activated Pre-assignment: TRUE
IN	boEnIChannel	BOOL	TRUE: I component is activated Pre-assignment: TRUE
IN	boEnDChannel	BOOL	TRUE: D component is activated Pre-assignment: FALSE
IN	boControlMode	BOOL	TRUE: D component is effective in the actual value branch (after smoothing) and not in the system deviation Only active, if the D component is activated Pre-assignment: FALSE
IN	r32DeadBand	REAL	[-] Width of the deadband, for *r32DeadBand* = 0.0, the deadband is deactivated. Pre-assignment: 0.0
IN	r32ActValueFactor	REAL	[-] Scaling factor of the actual value quantity Pre-assignment: 1.0
IN	r32ActValueOffset	REAL	[-] Offset of the actual value quantity Pre-assignment: 0.0
IN	r32OutValueFactor	REAL	[-] Scaling factor of the output quantity Pre-assignment: 1.0
IN	r32OutValueOffset	REAL	[-] Offset of the output quantity Pre-assignment: 0.0

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Functionality

The function block is used to implement a universal PID controller that can be switched over to the P controller or I controller modes. The D component can either be calculated in the actual value branch or in the system deviation.

The integrator can be set using the inputs *intSetValue* (setting value) and *setIntChannel*. In addition, the instantaneous value of the integrator can be held using *holdIntChannel* (P controller). Internally, the I component is limited, based on the controller output limits (“anti-windup”). The integrator is set to zero at switch-on.

All of the block parameters can be changed in operation.

The scaled and smoothed actual value (*actualValue*) is subtracted from the setpoint (*setpointValue*) corresponding to the following equation:

$$deviationValue = \left((actualValue \cdot r32ActValueFactor) + r32ActValueOffset\right) \cdot \left(1 - e^{-\frac{t}{T}}\right) - setpointValue$$

The result is the system deviation *deviationValue* (after the deadband) and is then multiplied with the proportional coefficient *propGain* that can be set. The result is transferred to the integrator – and optionally to the D component. The settable integral time *integTime* as well as the time constant of the D component *derivTime* (optional) define the controller behavior.

Table 4-3: Calculating the control components

Component	Calculation
P component	$pOutValue[n] = propGain \cdot deviationValue[n]$
I component	$iOutValue[n] = iOutValue[n-1] + \frac{pOutValue[n] + pOutValue[n-1]}{2} \cdot \frac{T_{Zyklus}}{integTime}$ T_{cycle}: Sampling time The integrator includes "anti-windup" functionality, i.e. if the controller output is limited, the integrator cannot continue to integrate in the same direction. The inputs *setIntChannel* and *holdIntchannel* can be used to set the integrator (*intSetValue*) – or manually hold it. The integrator is set to *zero* using *reset* or with a rising edge for *enable*.
D component (if activated)	$dOutValue[n] = (pOutValue[n] - pOutValue[n-1]) \cdot \frac{derivTime}{T_{cycle}}$ T_{cycle}: Sampling time

The controller output is first scaled and is then limited.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Warnings

Warnings are indicated by the status of the outputs *error* and *errorID*: *error* = FALSE and *errorID* <> 16#0000_0000.

ErrorID	Description
16#0000_0000	No warning
16#0110_4002	Invalid parameter, it is internally limited *derivTime* < *Tcycle,* *integTime* < *Tcycle / 2*

Errors

Error messages are indicated by the status of the outputs *error* and *errorID*: *error* = TRUE and *errorID* <> 16#0000_0000.

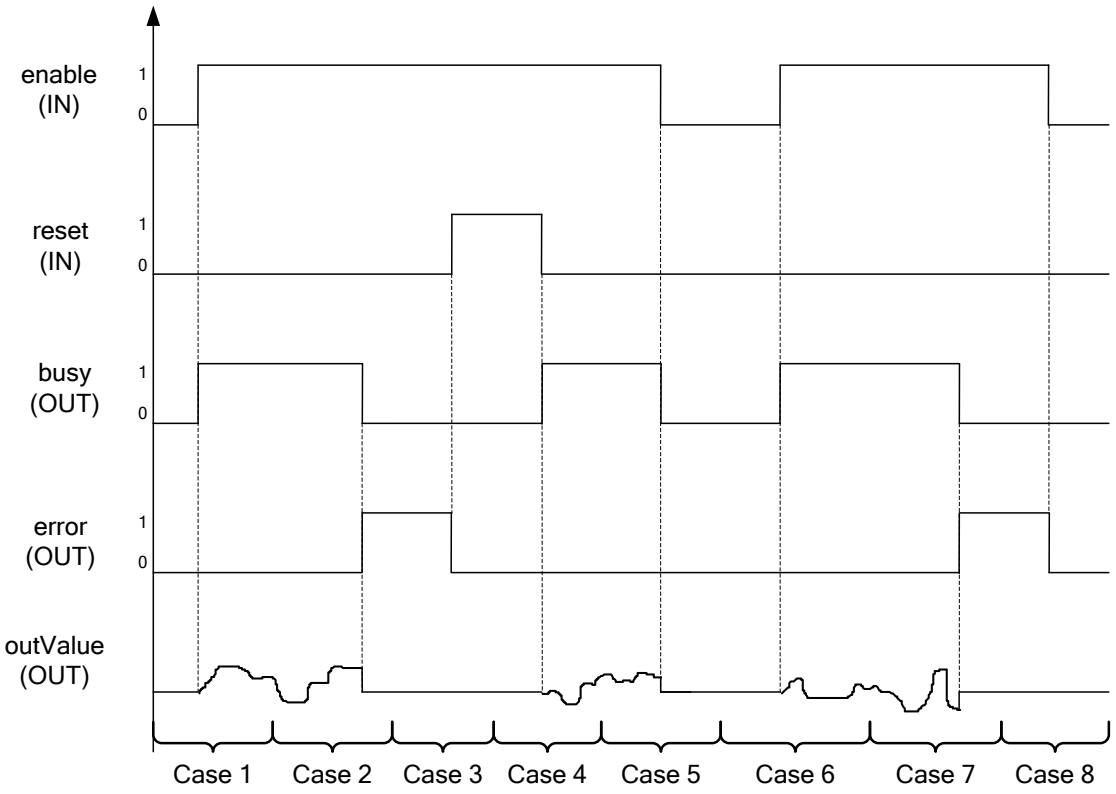
ErrorID	Description
16#0000_0000	No error
16#0110_8001	The function block is called in an incorrect task.
16#0110_8005	Invalid parameter *upperlimit* < *lowerlimit*

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Signal diagram

Fig. 4-10

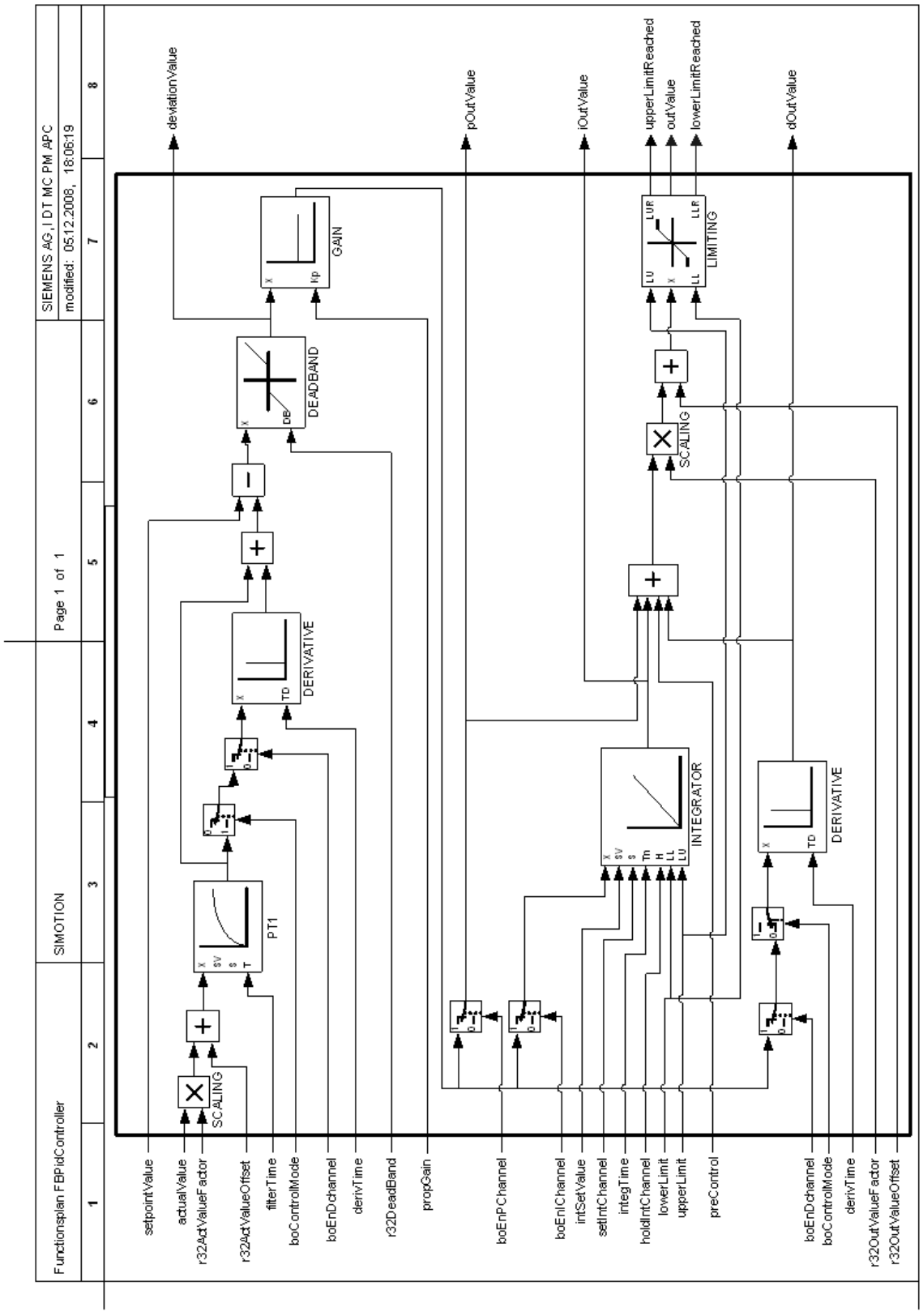


- Case 1, 6 – Blocked is switched-in with *enable* = *TRUE*. Output *busy* = *TRUE*, *outValue* is activated.
- Case 2, 7 – The error has been detected, *error* = *TRUE*, *busy* = *FALSE*, *outValue* = *0.0* – up to reset = TRUE or restart.
- Case 3 – Block is reset – as long as *reset* = *TRUE*. *Error* = *FALSE*.
- Case 4 – Block is reset and *reset* = *FALSE*, output *busy* = *TRUE*, *outValue* is activated.
- Case 5 – Block is deactivated with *enable* = *FALSE* while *busy* = *TRUE*. Busy changes to FALSE, *outValue = 0.0.*
- Case 8 – Block is deactivated with *enable* = *FALSE* while *error* = *TRUE*. Error changes to FALSE.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Function diagram



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

4.11 FBTechnologyController – Technology controller

Technology controller based on a PID controller for use as tension or dancer roll position controller in closed-loop winder or material web controls.

Block name

FBTechnologyController

Task

The function block includes the following functionality:

- Setpoint ramp-function generator with setting function
- PID controller
- Ramp function as controller output limiting
- Kp adaptation
- D component can be optionally switched-in in the actual value branch

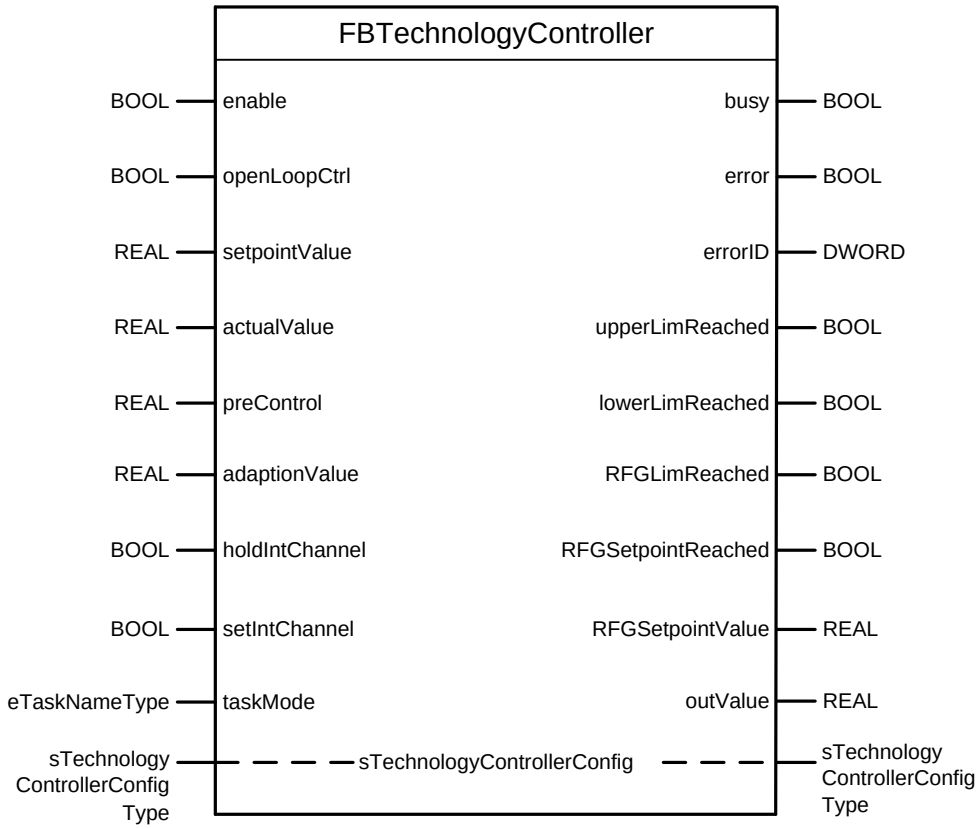
Integration in the run-time system

The function block must be called in a synchronous task (Servo, IPO, IPO_2 or Timer).

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Graphic representation of the block



Block parameters

Input parameters

Name	Data type	Initial value	Description
enable	BOOL	FALSE	FALSE -> TRUE: Ramp-function generators are started, PID controller is active (if not in the *openLoop* mode) TRUE -> FALSE: Setpoint is integrated to the actual value, the controller output is limited and then FB deactivated.
openLoopCtrl	BOOL	FALSE	FALSE: PID controller is active TRUE: PID controller and RFG limiting are inactive. Only the setpoint RFG is used.
setpointValue	REAL	0.0	[Z] / [LU] / [%] Setpoint signal
actualValue	REAL	0.0	[Z] / [LU] / [%] Actual value signal
preControl	REAL	0.0	[rpTU] or [°/TU] pre-control signal
adaptionValue	REAL	0.0	[-] Signal for Kp adaptation The units depend on the interconnection.
holdIntChannel	BOOL	FALSE	TRUE: I component is frozen
setIntChannel	BOOL	FALSE	TRUE: set I component

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

taskMode	eTaskName Type	IPO_ SYNCHRONO US_TASK	Specification of the task in which the block is called. SERVO_SYNCHRONOUS_TASK IPO_SYNCHRONOUS_TASK IPO_SYNCHRONOUS_TASK_2 TIMER_INTERRUPT_TASK

Input/output parameters

Name	Data type	Initial value	Description
sTechnology Controller Config	sTechnology Controller ConfigType	-	Configuration data of the technology controller. Details on this structure are provided in the description of the type definitions.

Output parameters

Name	Data type	Initial value	Description
busy	BOOL	FALSE	FALSE: The function block is deactivated TRUE: The function block is activated. The controller output or the ramped setpoint (in case of *openLoopCtrl = TRUE*) is not zero.
error	BOOL	FALSE	FALSE: No error TRUE: An error has occurred, refer to error identification
errorID	DWORD	16#0000_0000	Error identification, refer to error messages errorID <= 16#000_8000: warning errorID >= 16#0000_8000: error
upperLimReached	BOOL	FALSE	TRUE: Output is limited, upper limit
lowerLimReached	BOOL	FALSE	TRUE: Output is limited, lower limit
RFGLimReached	BOOL	FALSE	TRUE: Output ramp-function generator has reached the limit (= 100%).
RFGSetpoint Reached	BOOL	FALSE	TRUE: Setpoint has been reached, the ramp-function generator output corresponds to the setpoint.
RFGSetpointValue	REAL	0.0	[Z] / [LU] / [%] Setpoint after the ramp-function generator
outValue	REAL	0.0	[LU/TU] or [Z] controller output after scaling and limiting. The units are dependent on the interconnection.

Note

The calculations in this function block are independent of the physical units.

[LU] Length unit, e.g. m
[Z] Tension, force, e.g. N

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Data types and structures

Name of the structure	Contents
sPidDataType	Configuration data of the PID controller Declared in: LConLib.fConvTools
sTechnologyController ConfigType	Configuration data of the technology controller Declared in: LConLib. fConvTools

sPidDataType

The following type declaration applies:

I/O	Element	Data type	Description
IN	boEnPChannel	BOOL	TRUE: P component is activated Pre-assignment: TRUE
IN	boEnIChannel	BOOL	TRUE: I component is activated Pre-assignment: TRUE
IN	boEnDChannel	BOOL	TRUE: D component is activated Pre-assignment: FALSE
IN	boControlMode	BOOL	TRUE: D component is effective in the actual value branch (after smoothing) and not in the system deviation Only active, if the D component is activated Pre-assignment: FALSE
IN	r32DeadBand	REAL	[-] Width of the deadband, for *r32DeadBand* = 0.0, the deadband is deactivated. Pre-assignment: 0.0
IN	r32ActValueFactor	REAL	[-] Scaling factor of the actual value quantity Pre-assignment: 1.0
IN	r32ActValueOffset	REAL	[-] Offset of the actual value quantity Pre-assignment: 0.0
IN	r32OutValueFactor	REAL	[-] Scaling factor of the output quantity Pre-assignment: 1.0
IN	r32OutValueOffset	REAL	[-] Offset of the output quantity Pre-assignment: 0.0

sTechnologyControllerConfigType

The following type definition applies:

I/O	Element	Data type	Description
IN	r32X1	REAL	[-] x coordinate for Kp adaptation for Kp1
IN	r32Kp1	REAL	[-] Kp value for *r32X1*. If the Kp adaptation is deactivated, then this parameter is used as Kp value.
IN	r32X2	REAL	[-] x coordinate for Kp2 If *r32X1* = *r32X2*, then the Kp adaptation is deactivated.
IN	r32Kp2	REAL	[-] second Kp value, assigned corresponding to *r32X2*.
OUT	r32EffectiveKp	REAL	[-] effective controller gain after Kp adaptation
IN	r32LimRampUpTime	REAL	[ms] ramp-up time of the controller output limiting. This is internally limited to the cycle time as lower value.
IN	r32LimRampDownTime	REAL	[ms] ramp-down time of the controller output limiting.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

I/O	Element	Data type	Description
			This is internally limited to the cycle time as lower limit.
IN	r32SetNominalValue	REAL	[Z] / [LU] / [%] setpoint RFG, reference value for ramp-up/ramp-down time.
IN	r32SetRampUpTime	REAL	[ms] setpoint RFG ramp-up time This is internally limited to the cycle time as lower limit.
IN	r32SetRampDownTime	REAL	[ms] setpoint RFG ramp-down time This is internally limited to the cycle time as lower limit.
IN	r32IntegTime	REAL	[ms] time constant for I component This is internally limited to the cycle time / 2 as lower limit.
IN	r32DerivTime	REAL	[ms] time constant for D component This is internally limited to the cycle time as lower limit.
IN	r32FilterTime	REAL	[ms] time constant for actual value smoothing This is internally limited to the cycle time as lower limit. If *r32FilterTime* = 0.0, then smoothing is deactivated.
IN	r32IntSetValue	REAL	[LU/TU] / [Z] integrator setting value
IN	sPidData	sPidDataType	PID controller configuration data, refer to the type description
IN	r32UpperLimit	REAL	[LU/TU] / [Z] controller output, upper limit value
IN	r32LowerLimit	REAL	[LU/TU] / [Z] controller output, lower limit value. Must be less than *r32UpperLimit*.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Functionality

The block is enabled using the *enable* input; as a consequence, the setpoint ramp-function generator and the limiting ramp-function generator are started with their respective ramp-up times and the PID controller is activated. In addition, for a rising edge, the setpoint ramp-function generator is set to the actual value that is present (*actualValue*). This guarantees a jerk-free and smooth transition into the tension controlled mode.

When the edge at the enable input falls, the limiting ramp-function generator is ramped back to zero and therefore the controller output is limited. This results in a bumpless controller shutdown. As soon as the *outValue* output is zero and *enable* = FALSE, the block is deactivated and the output *busy* is set to FALSE. The setpoint is internally ramped to the actual value during the switch-off phase.

If the input *openLoopCtrl* = TRUE, then the block operates in the open control circuit mode. As a consequence, the setpoint channel remains active and the controller or the output limits remain inactive. The output *busy* stays *TRUE* as long as the setpoint ramp output is not zero. The input *actualValue* is 0.0.

Setpoint ramp

The setpoint ramp-function generator is used to avoid setpoint steps in the setpoint channel. The setpoint ramp-function generator is configured using parameter *r32SetRampDownTime* , *r32SetRampUpTime, and r32SetNominalValue*, whereby *r32SetNominalValue* defines the reference value for the ramp-up or ramp-down time. The function is always active, if *enable* = *TRUE*.

If *openLoopCtrl* = FALSE*, enable* = FALSE and *busy* = TRUE (switch-off phase), then the setpoint is internally set to the actual value.

The ramp times are internally limited to the cycle time.

Limit ramp

The limit ramp is only active, if the technology controller is used as closed control loop. When switching on the controller with *enable* = TRUE, the limits are ramped to 100% using the parameterized ramp-up time (*r32LimRampUpTime*). The output limit (*r32UpperLimit* and *r32LowerLimit*) is therefore completely effective.

When switching off (*enable* = FALSE), the controller limits are ramped back to 0% (*r32LimRampDownTime*), to permit bumpless transition into the non-controlled mode. The controller is deactivated once zero has been reached.

The times are internally limited to the cycle time.

Controller function

Controller setpoint is provided via the *setValue*, the actual value via the *actValu*e input. The setpoint channel includes a settable setpoint ramp-function generator, via which the setpoint is routed; the effective setpoint can be monitored at output *RFGSetValue*. The control algorithm used corresponds to that described in block FBPidController.

Kp adaptation

A linear adaptation is carried out as Kp adaptation. The interpolation straight line is defined by the value tuple ***(r32X1, r32Kp1)*** and ***(r32X2, r32Kp2)***.

1st case: (adaptionValue < r32X1) or (r32X1 = r32X2):

r32EffectiveKp = r32Kp1

2nd case: r32X1 <= adaptionValue <= r32X2:

$$r32EffectiveKp = r32Kp1 + \frac{(adaptionValue - r32X1)(r32Kp2 - r32Kp1)}{(r32X2 - r32X1)}$$

3rd case: adaptionValue > r32X2:

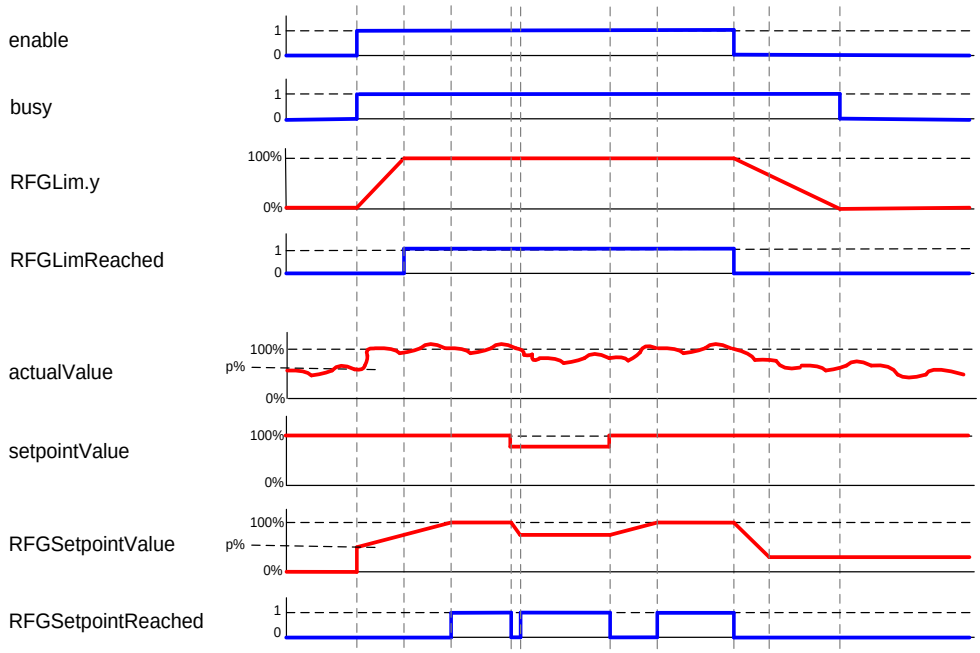
r32EffectiveKp = r32Kp2

r32Kp1 is used as effective Kp if the Kp adaptation is deactivated.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Graphic representation of the function



Warnings

Warnings are indicated by the status of the outputs *error* and *errorID*: *error* = FALSE and *errorID* <> 16#0000_0000.

ErrorID	Description
16#0000_0000	No warning
16#0208_4002	Invalid parameter, internal limitation *derivTime* < *Tcycle*, *integTime* < *Tcycle / 2*

Errors

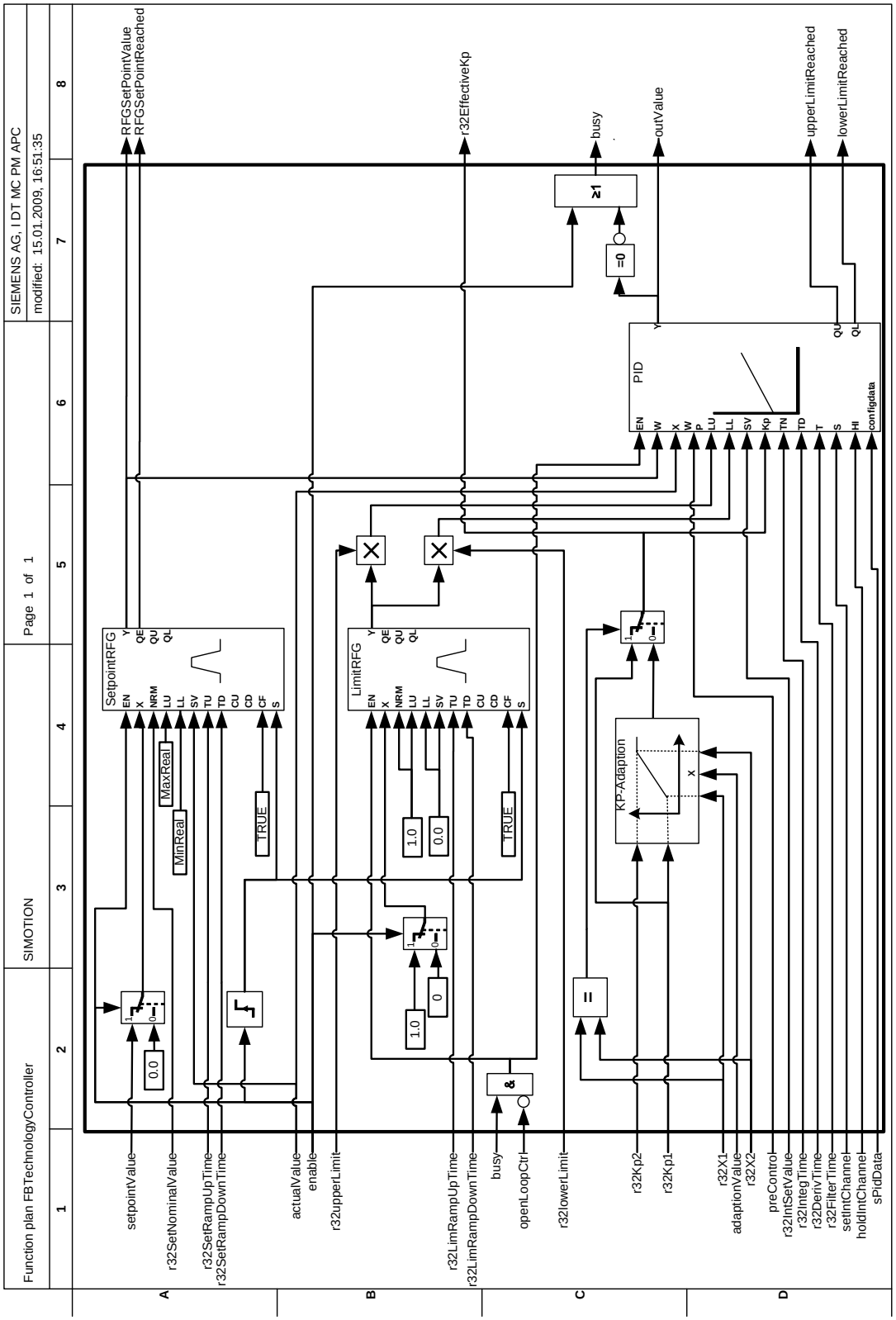
Error messages are indicated by the status of the outputs *error* and *errorID*: *error* = TRUE and *errorID* <> 16#0000_0000.

ErrorID	Description
16#0000_0000	No error
16#0208_8001	The function block is called in an incorrect task.
16#0208_8005	Invalid parameter: *sTechnologyControllerConfig.r32Upperlimit* <= *sTechnologyControllerConfig.r32Lowerlimit* *sTechnologyControllerConfig.r32SetNominalValue* <= *0.0*

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Function diagram



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

5 Winding functions

The description of the function block FBWinder can be found in a sperate document:

https://support.industry.siemens.com/cs/de/en/view/35818687

5.1 FBDiameterCalculation – Diameter calculator

Function block to calculate the actual diameter of an axial winder.

Block name

FBDiameterCalculation

Task

This function block calculates the actual diameter of the roll based on the behavior of the winding drive and line drive. One of 5 modes can be selected for the calculation. The result of the diameter calculation is subject to a plausibility check.

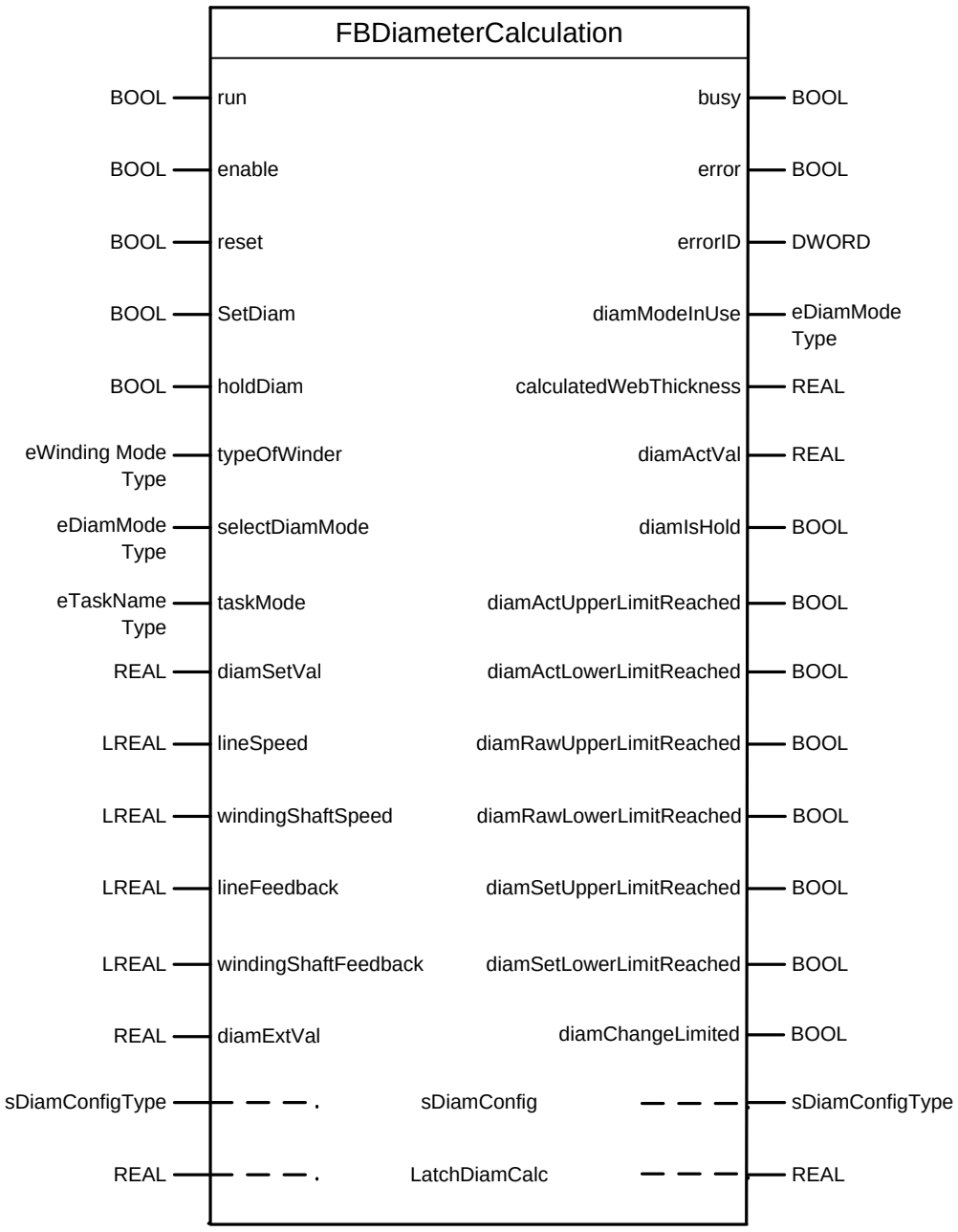
Integration in the run-time system

The function block must be called in a synchronous task (Servo, IPO, IPO_2 or Timer).

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Block parameters

Input parameters

Name	Data type	Initial value	Description
run	BOOL	TRUE	FALSE: function block is deactivated The block sets the output signals to “secure” values and executes a limitation of the output value *diamActVal*. The limitation is done between *sDiamConfid.diamMin* and *sDiamConfig.diamMax*. TRUE: function block is activated
enable	BOOL	FALSE	FALSE: The diameter calculator is deactivated (acts like holdDiam) TRUE: The diameter calculator is activated. The calculation mode is assumed and internal stored with a rising edge.
reset	BOOL	FALSE	TRUE: Resets all of the variables in the FB (also filter)
setDiam	BOOL	FALSE	TRUE: Sets the diameter to a specified value The value is specified in *sDiamConfig.diamSetVal*
holdDiam	BOOL	FALSE	TRUE: Holds the actual diameter
typeOfWinder	eWinding ModeType	REWIND_ FROM_ ABOVE	Winder type: REWIND_FROM_ABOVE REWIND_FROM_BELOW UNWIND_FROM_ABOVE UNWIND_FROM_BELOW Only used with *selectDiamMode* = DIAM_THICKNESS_ADDITION.
selectDiamMode	eDiam ModeType	DIAM_ CALC	Selects the diameter calculation mode DIAM_CALC: V/N calculation DIAM_CALC_INTEGRAL: Integrating up from the winding speed and web velocity DIAM_CALC_POSITION: Measures the position of the winding angle and material web DIAM_THICKNESS_ADDITION: The material thickness is continually added DIAM_EXT: The external diameter value is used Can not be changed while *enable* = TRUE.
taskMode	eTaskName Type	IPO_ SYNCHRONO US_ TASK	Task in which the FB is called: SERVO_SYNCHRONOUS_TASK: Servo Synchronous Task IPO_SYNCHRONOUS_TASK: Ipo Synchronous Task IPO_SYNCHRONOUS_TASK_2: Ipo Synchronous Task 2 TIMER_INTERRUPT_TASK: in a Timer Interrupt Task
diamSetVal	REAL	0.0	[LU] Setting value of the diameter calculator Assumption when *setDiam* = TRUE, as long as the block is deactivated.
lineSpeed	LREAL	0.0	[LU / s] Web/machine velocity Generally, the web velocity is the setpoint velocity of the machine and the velocity with which the main drive moves the material web. In exceptional cases, or with a deep understanding of control technology, it is possible to use the actual value from a tachometer measuring the web speed, etc.
windingShaftSpeed	LREAL	0.0	[r / s] Speed of the winder shaft The winder motor is fed from a variable-speed drive that has an actual value encoder that has sufficient resolution. This actual value signal is converted to the roll shaft and is used to calculate the diameter.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Name	Data type	Initial value	Description
lineFeedback	LREAL	0.0	[Inc] Value of the counter variable of the web – position of the web
windingShaftFeedback	LREAL	0.0	[Inc] Value of the counter variable of the winder shaft – position (angle) of the winder shaft
diamExtVal	REAL	0.0	[-] External diameter value, can be scaled by the characteristics, if necessary.

Input / output parameters

All parameters that are contained in sDiamConfig are input parameters.

Name	Data type	Initial value	Description
latchDiamCalc	REAL	0.0	[LU] Retain variable, the calculated diameter is saved
sDiamConfig	sDiam Config Type		Structure for diameter calculator configuration data Details about this structure are provided in the description of the type definitions.

Output Parameters

Name	Data type	Initial value	Description
busy	BOOL	FALSE	FALSE: Function block is deactivated, values will not be evaluated TRUE: Function block is activated
error	BOOL	FALSE	FALSE: No error TRUE: An error has occurred, refer to error identification
errorID	DWORD	16#0000_0000	Error identification, refer to error messages errorID <= 16#000_8000: warning errorID >= 16#0000_8000: error
diamModeInUse	eDiam ModeType	-	Actual diameter calculation mode
calculatedWebThickness	REAL	0.0	[LU] Calculated value of the web thickness only for DIAM_CALC_INTEGRAL and DIAM_CALC_POSITION mode. Otherwise the output is set to 0.0.
diamActVal	REAL	0.0	[LU] Actual diameter value
diamIsHold	BOOL	FALSE	FALSE: Diameter calculator is enabled TRUE: Diameter calculator is in the stop condition
diamActUpperLimit Reached	BOOL	FALSE	TRUE: Upper limit of the actual diameter value reached
diamActLowerLimit Reached	BOOL	FALSE	TRUE: Lower limit of the actual diameter value reached
diamRawUpperLimit Reached	BOOL	FALSE	TRUE: Upper limit of the actual raw diameter value is reached
diamRawLowerLimit Reached	BOOL	FALSE	TRUE: Lower limit of the actual raw diameter value is reached
diamSetUpperLimit Reached	BOOL	FALSE	TRUE: Upper limit of the diameter setpoint is reached
diamSetLowerLimit Reached	BOOL	FALSE	TRUE: Lower limit of the diameter setpoint is reached
diamChangeLimited	BOOL	FALSE	TRUE: Diameter change rate is limited.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Note

In principle, the calculations made in this function block are independent of any physical units. The input parameter determines the units of the output parameters. The internal calculations carry-out specific related conversions

[LU]	Length unit, e.g. m
[TU]	Time unit of the *line-/winderAxis*
[Inc]	Position signal of the winding and the material web axis
[V]	Web velocity, derived from LU/min
[n]	Speed, e.g. rpm

As far as the physical units of smaller length units and conversion of velocities, conversions are made into LU/s or rev/s.

Functionality

The diameter calculator determines the actual diameter value (*diamActVal*) of the roll being wound (or unwound) using the input quantities:

- Web velocity and position
- Shaft speed of the mandrel or angle
- Web thickness

Calculation modes

The function block implements the following calculation modes:

Table 5-1: Diameter calculation types

Calculation mode	Description
DIAM_THICKNESS_ADDITION	Integrates the web thickness
DIAM_CALC	Calculates the diameter based on the ratio between the web velocity and winder velocity or speed (speed)
DIAM_EXT	Senses the diameter using a sensor
DIAM_CALC_INTEGRAL	Calculates the diameter from the ratio between the winder angle and web length by integrating the winding speed and the web velocity.
DIAM_CALC_POSITION	Calculates the diameter using the ratio between the winder angle and web length by measuring the winder position (angle) and the web length.

Parameter *selectDiamMode* specifies the calculation modes.
At the first run through or after a reset, the specified mode is accepted and is issued at output parameter *diamModeInUse* for the user.

The diameter is calculated in two steps:

- The raw diameter value is calculated
- The raw diameter value is post-conditioned (limited, verified and smoothed)

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Calculating the raw diameter value

The following structure parameters must always be made available independent of the selected calculation mode and the input parameters with which the FB is directly supplied.

Table 5-2: Structure parameters

Name	Description
latchDiamCalc	[LU] Retain variable old/ previous diameter
sDiamConfig	
.r32MinLineSpeed	[LU / s] velocity limit of line velocity for activation of the diameter calculator. If *lineSpeed* < *r32MinLineSpeed* the diameter is hold.
.r32MinWinderSpeed	[r / s] velocity limit of winder speed for activation of the diameter calculator. If *windingShaftSpeed* < *r32MinWindingShaftSpeed* the diameter s hold.
.r32DiamMax	[LU] Maximum diameter value
.r32DiamMin	[LU] Minimum diameter value
.r32WebThickness	[LU] Thickness of the web, used for the limit of the diameter rising and used in calculation mode DIAM_THICKNESS_ADDITION
.r32AdjustWebThickness	[-] Factor to adjust rampupTime / rampdownTime of the ramp-function generator
.r32SmoothingTime LayerJump	[ms] Layer jump smoothing The layer jumps are smoothed with this smoothing factor dependent on diamActVal.
.r32SelectIntegration Direction	Integration direction of the ramp-function generator DONTCARE: Can integrate in both directions INCREASING: Only increasing (away from 0.0) DECREASING: Only decreasing (in direction of 0.0)

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

DIAM_THICKNESS_ADDITION

This mode calculates the diameter by continually adding the web thickness that is added or removed from the roll at each sampling time.
The diameter change is 2 x web thickness (h) x speed of the mandrel (n). Starting from an initial diameter, the actual diameter value is formed from this change through numerical integration.
Depending on the winder type e*TypeOfWinder* the material thickness is either added or subtracted.

Table 5-3: Parameter list, material thickness integration

Name	Description
sDiamConfig	Structure for the diameter calculator configuration data
.eTypeOfWinder	Winder type UNWIND_FROM_ABOVE UNWIND_FROM_BELOW REWIND_FROM_ABOVE REWIND_FROM_BELOW

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

DIAM_CALC

The diameter is obtained by dividing the web quantity by the angular quantity whereby the following applies:

$$v = R \cdot \omega, \ \ with\, D = \frac{v}{\pi \cdot n}$$

In order that the associated quantities are divided at the same instant of detection, it is also possible to slightly delay the web velocity (v) and/or the winder speed (n) using a PT1 element.

Table 5-4: Parameter list V/n diameter calculation

Name	Description
sDiamConfig	Structure for the diameter calculator configuration data
.r32LineBalancingTime	[ms] Symmetrizing time This parameter correlates between the velocity and speed. It compensates the offset in time of lagging speed n of the winding shaft with respect to the machine velocity v.
.r32WindingShaft BalancingTime	[ms] Symmetrizing time This parameter correlates between the velocity and speed. It compensates the offset in time of lagging machine velocity v with respect to the speed n of the winding shaft.

DIAM_EXT

In this particular mode, the diameter is externally sensed using an analog sensor. The numerical sensor value is assigned to parameter *diamExtVal*. In the diameter calculator, this value forms the abscissa value of a characteristic block (refer to *FBPolygonA*). The user should parameterize this characteristic block so that its output value *ordinateOutput* is the actual diameter.

By setting the DiamConfig parameter u16DiamExtCurveEntryNum to zero, the functionblock FBPolygonA won't be calculated and the value assigned to diamExtVal will be taken as diameter.

Table 5-5: Parameter list, external diameter

Name	Description
diamExtVal	[-] External diameter value, when required, can be scaled and linearized using the characteristic.
sDiamConfig	Configuration data of the diameter calculator
.u16DiamExtCurve EntryNum	[-] Number of grid points for the polygon calculation [2 – LCONLIB_MAX_IDX_A+1]
.r32DiamExtCurveAbscissa	[-] Abscissa values for the diameter characteristic to linearize the external diameter signal (maximum LCONLIB_MAX_IDX_A+1 x-y points).
.r32DiamExtCurveOrdinate	[LU] Ordinate values for the diameter characteristic to linearize the external diameter signal (maximum LCONLIB_MAX_IDX_A+1 x-y points).

DIAM_CALC_INTEGRAL

These integral diameter calculation techniques are also based on the division of the web quantity by the angular quantity. However, quantities v and ω are integrated over a number of roll revolutions that can be specified (*sDiamConfig.r32MeasuringRevolutionCount*). This is to minimize the disturbing effect of processing just individual values. The following applies:

$$s = R \cdot \varphi, \quad with\ D = 2 \cdot \left(\frac{s}{\varphi} \right)$$

If the number of revolutions of the winding shaft - specified by parameter *sDiamConfig.r32MeasuringRevolutionCount* – is reached, then the integrators are reset. The calculated diameter value is then further processed.
In order that the associated quantities are divided at the same instant of detection it is also possible to slightly delay the web velocity signal (v) using a PT1 element.

In addition in every measuring interval the actual material thickness (*r32CalculatedWebThickness*) is calculated, e.g. for a rest length calculation.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Table 5-6: Parameter list, integral diameter calculation

Name	Description
sDiamConfig	Structure for the configuration data of the diameter calculator
.r32LineBalancingTime	[ms] Symmetrizing time This parameter correlates between the velocity and speed. It compensates the offset in time of lagging speed n of the winding shaft with respect to the machine velocity v.
.r32WindingShaft BalancingTime	[ms] Symmetrizing time This parameter correlates between the velocity and speed. It compensates the offset in time of lagging machine velocity v with respect to the speed n of the winding shaft.
.r32MeasuringRevolution Count	[-] Number of winding shaft revolutions that define the measuring and calculation interval
.r32FirstDiameterOutput	[-] Number of winding shaft revolutions after which the first calculated diameter is output.
.r32DiameterOutputRate	[-] Number of winding shaft revolutions after which the next calculated diameter is output.

DIAM_CALC_POSITION

This diameter calculation technique directly intervenes at the position of the web and the roll being wound or unwound. In this case, the position is fed back into the FB via the input parameter *lineFeedback* and *windingShaftFeedback* there (in the FB) the difference to the previous cycle is measured. Overflows are intercepted using modulo limits that should be entered. Positions are evaluated and weighted using the scaling factors *sDiamConfig.r64LineFeedbackScal* and *sDiamConfig.-r64WindingShaftFeedbackScal*.

sDiamConfig.r64LineFeedbackScal corresponds to: 1 [LU]

sDiamConfig.r64WindingShaftFeedbackScal corresponds to: 1 [rev]

$$s = R \cdot \varphi, \quad with\ D = 2 \cdot \left(\frac{s}{\varphi} \right)$$

If the number revolutions of the winding shaft – specified by parameter *sDiamConfig.r32MeasuringRevolutionCount* – is reached the differential value is reset and the calculated diameter value is further processed. In order to be able to identify overflows, the change of the feedback signal per cycle must be less than half of the difference of the modulo limits.

$$\Delta Feedback < \frac{ModLength}{4}$$

In addition in every measuring interval the actual material thickness (*r32CalculatedWebThickness*) is calculated, e.g. for a rest length calculation.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Table 5-7: Parameter list, position-based diameter calculation

Name	Description
sDiamConfig	Structure for diameter configuration data
.r32MeasuringRevolution Count	[-] Number of winding shaft revolutions that define the measuring and calculation interval
.r32FirstDiameterOutput	[-] Number of winding shaft revolutions after which the first calculated diameter is output.
.r32DiameterOutputRate	[-] Number of winding shaft revolutions after which the next calculated diameter is output.
.r64WindingShaftFeedback Scal	[Inc/rev] Number of increments that correspond to 1 [rev]
.r64LineFeedbackScal	[Inc/LU] Number of increments that correspond to 1 [LU]
.r64LineModLength	[Inc] Number of increments for the modulo length of the LineFeedback Signals
.r64ShaftModLength	[Inc] Number of increments for the modulo length of the WindingShaftFeedback Signals

Floating diameter calculation in modes DIAM_CALC_INTEGRAL, DIAM_CALC_POSITION

The floating diameter calculation can be configured by the following configuration parameter:

- r32MeasuringRevolutionCount
- r32FirstDiameterOutput
- r32DiameterOutputRate

After rising edge of input “enable” resp. after a change of the diameter calculation mode the first calculated diameter will be output after the number of revolutions configured in “r32FirstDiameterOutput”.

After the first diameter output the calculated diameter will be refreshed cylic after the number of revolutions configured in “r32DiameterOutputRate”.

The calculated diameter will be averaged over the number of revolutions configured in “r32MeasuringRevolutionCount”, as long as the the parameterized number of revolutions has been passed already.

The averaging of a specific number or revolutions is determinded by the depth (array length) of an internal ring buffer. Inside the ring buffer the position deltas of line and winder axis will be entered. The buffer depth is calculated based on the following formula:

$$buffer - depth = \frac{r32MeasuringRevolutionCount}{r32DiameterOutputRate}$$

Is the result of the calculation a non natural number the value will be rounded to next natural number (N = {1, 2,3, …}).Therby the the effective averaging over a specific number of revolutions can differ to the configured value in “r32MeasuringRevolutionCount” (compare example 3).

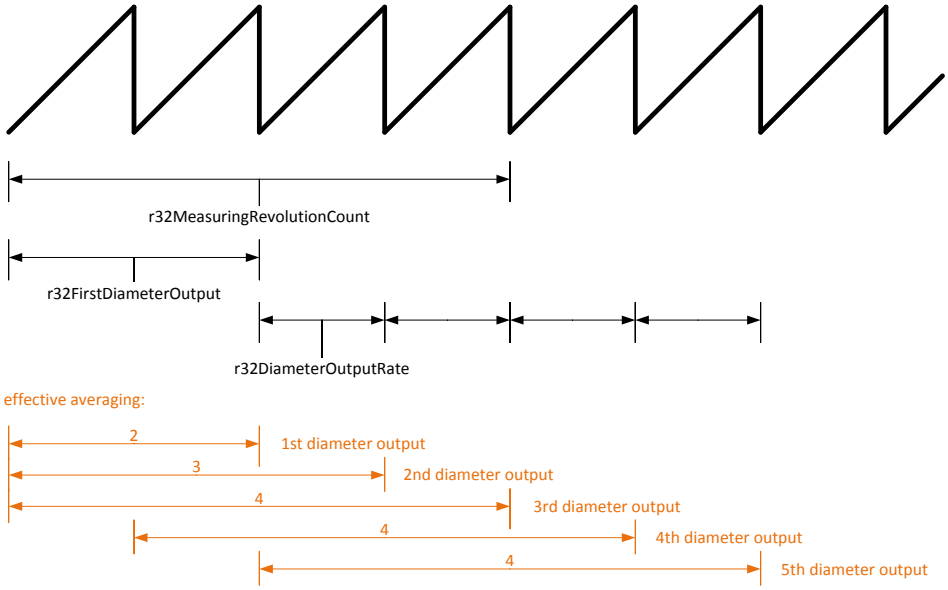
The following sketeches show the interacting of the configuration parameter. In the function block a modulo cycle from 0° to 360° is recreated. The configuration of the technology object is independend of that.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

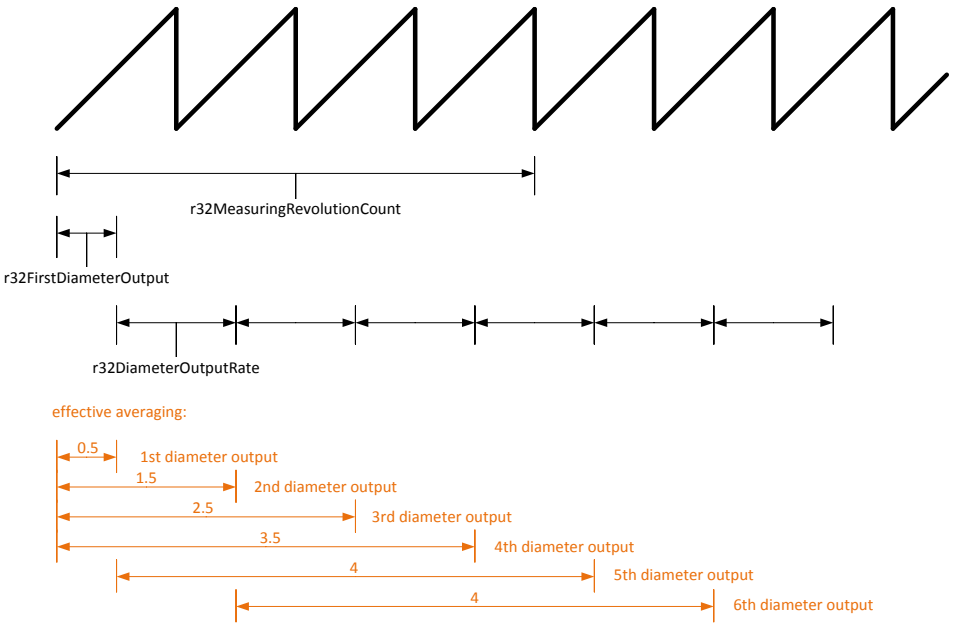
Example 1:

r32MeasuringRevolutionCount := 4
r32DiameterOutputRate := 1
r32FirstDiameterOutput := 2



Example 2:

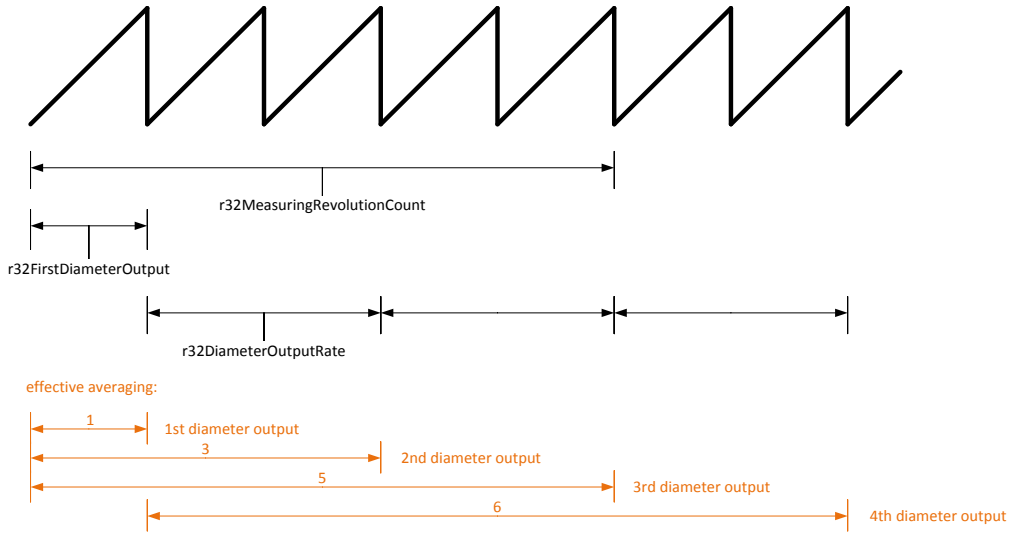
r32MeasuringRevolutionCount := 4
r32DiameterOutputRate := 1
r32FirstDiameterOutput := 0.5



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Example 3:

r32MeasuringRevolutionCount := 5
r32DiameterOutputRate := 2
r32FirstDiameterOutput := 1



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Post processing the raw diameter value

Limiting the raw diameter value:
After the calculation using one of the five above mentioned calculating modes, the raw diameter is limited to the minimum diameter and maximum diameter values:

r32DiamMin ≤ *diamActRawVal* ≤ *r32DiamMax*

r32DiamMin, *r32DiamMax*: -> *sDiamConfig*

Keeping the limited raw diameter value constant:

It only makes sense to further process the raw diameter value if both the web velocity as well as the winder shaft speed (*lineSpeed*, *windingShaftSpeed*) exceed their specific minimum values (*sDiamConfig.r32MinLineSpeed*, *sDiamConfig.r32MinWinderSpeed*):

lineSpeed > *r32MinLineSpeed*

windingShaftSpeed > *r32MinWinderSpeed*

If one of these conditions is not fulfilled, then internally the diameter is held – it is not changed.
The same result – i.e the diameter is kept constant – is obtained when the input signal *holdDiam* at the function block is active.

Saving the limiting raw diameter value:

The raw diameter value is formed and retentively saved at each FB call in the calculation modes *DIAM_THICKNESS_ADDITION*, *DIAM_CALC* and *DIAM_EXT* ´(refer to the *latchDiamCalc* parameter).
In the integrating calculation modes

DIAM_CALC_INTEGRAL as well as

DIAM_CALC_POSITION the raw diameter value is only saved at the end of the integration or end of the calculation. This means when the specified number of winding shaft revolutions has been reached.

Further processing the latchDiamCalc value:

The further processing of the *latchDiamCalc* value essentially comprises smoothing using an interpolator with a PT1 element connected at its output.

Interpolation modes

The following interpolation modes are implemented in the function block:

Table 5-8: Interpolation modes

Interpolation mode	**Description**
NO_INTERPOLATION	No interpolation
INTERPOLATION_1	Interpolation with ramp-function generator, available in every calculation mode. The material thickness is required for the plausibility check.
INTERPOLATION_2	Interpolation with ramp-function generator, only available in the modes DIAM_CALC_INTEGRAL and DIAM_CALC_POSITION.
INTERPOLATION_3	Interpolation without ramp-function generator, only available in the calculation modes DIAM_CALC_INTEGRAL and DIAM_CALC_POSITION. The material thickness is required for the plausibility check.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

The interpolation mode is selected using the *sDiamConfig.eInterpolationMode* parameter.

Each time it is called – if used – the ramp-function generator executes the following iteration:

$$Y_k = Y_{k-1} + Y_A$$

The value for the acceleration is thereby obtained:

$$Y_A = \left(\frac{NRM}{T_D} \right) \cdot T_A$$ when braking, and

$$Y_A = \left(\frac{NRM}{T_U} \right) \cdot T_A$$ when accelerating

T_A: = task cycle time

When considered from the physical perspective the acceleration values correspond to the diameter changes per unit time. The following applies:

$$\left(\frac{NRM}{T_D} \right) = \frac{\Delta D}{\Delta t}$$ as the diameter decreases

$$\left(\frac{NRM}{T_U} \right) = \frac{\Delta D}{\Delta t}$$ as the diameter increases

Converting Library

The diameter calculator either sets the normalization value to 1.0 (ramp-function generator is operational) or to 0.0 (ramp-function generator is inhibited).

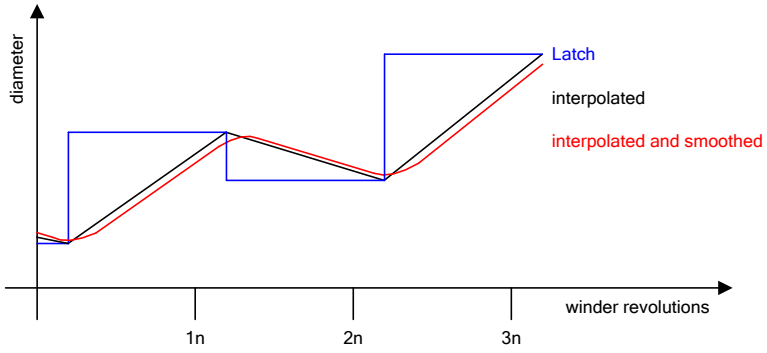
The other two ramp-function generator input parameters *TD* and *TU* are calculated by changing over the above specified formula:

$$T_U = T_D = \left(\frac{\Delta D}{\Delta t} \right)^{-1}$$

Table 5-9: Interpolation modes

Interpolation mode	Description
INTERPOLATION_1	In this mode, the differential quotient ΔD/Δt is calculated over the increase in thickness of the wound roll: $\frac{\Delta D}{\Delta t} = \frac{r32AdjustWebThickness \cdot r32WebThickness \cdot \omega}{2 \cdot \pi}$ $\frac{\Delta D}{\Delta t} = \frac{2 \cdot r32AdjustWebThickness \cdot r32WebThickness \cdot lineSpeed}{\pi \cdot diamActVal}$ The ramp-function generator input parameters *TD* and *TU* are calculated as follows: $TU = TD = \frac{NRM \cdot \pi \cdot diamActVal}{2 \cdot r32AdjustWebThickness \cdot r32WebThickness \cdot lineSpeed}$ This interpolation mode is available for all diameter calculation types.
INTERPOLATION_2	T_U or T_D are calculated as in the first mode, however, under the following conditions: • T_U (or T_D) is the time interval T_R between two reset pulses of the integrators. The value of T_R is obtained from the sum of the cycle times for the duration of the integration. • NRM is the change in diameter Δ*diamActVal* in this time interval. The value of Δ*diamActVal* is obtained from the difference between the actual and previous raw diameter value. These values could be immediately fed to the ramp-function generator. However, by setting *NRM* to 1.0 m or 0.0 m, we switch the complete ramp-function generator in or out – and must therefore execute the following calculation: $Y_A = \frac{T_A}{T_U} \cdot NRM = \frac{T_A}{\frac{T_R}{r32AdjustWebThickness \cdot \Delta diamActVal}} \cdot 1$ and therefore: $T_U = \frac{T_R}{r32AdjustWebThickness \cdot \Delta diamActVal}$ T_R is now obtained from the sum of the cycle times for the duration of the integration. This corresponds to the value from Δ*diamActVal* from the difference between the actual and previous raw diameter value. This means that the values for T_U and T_D can be precisely calculated. This interpolation mode is not available for the diameter calculation mode *DIAM_THICKNESS_ADDITION*, *DIAM_CALC*, *DIAM_EXT*.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Interpolation mode	Description
INTERPOLATION_3	In this mode, the interpolation is numerically realized using the *FCLinear* block. The interpolation is realized referred to the parameterized measurement interval between two measuring points based on the calculated diameter change at the measuring point. The calculated diameter change is limited by the configured material thickness, the maximum change is as follows: $\Delta \text{Diameter}_{max} = 2 * \text{Material thickness} * \text{Measurement revolutions}$  n: number of revolutions per calculated raw value This interpolation mode is not available for the diameter calculation modes DIAM_THICKNESS_ADDITION**,** DIAM_CALC**,** DIAM_EXT.

Note

With the calculation modes DIAM_CALC_INTEGRAL and DIAM_CALC_POSITION with the interpolation modes *INTERPOLATION_1* and *INTERPOLATION_2* after holding the diameter calculator and with the web thickness precisely entered, the actual diameter is no longer reached. For these modes, the thickness should be manipulated in a specific fashion using the factor *sDiamConfigType.r32AdjustWebThickness* in order to correct the diameter value.

The *output diamChangeLimited* will be set, if the change rate is limited.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Warnings

Warnings are indicated by the status of the outputs *error* and *errorID*: *error* = FALSE and *errorID* <> 16#0000_0000.

ErrorID	Description
16#0000_0000	No warning
16#0201_4002	The latchDiamCalc variable contains a value that lies outside the specified limits. It is probable that either no diameter has been set or an incorrect diameter has been entered.
16#0201_4015	The table of entered values is not monotonically (steadily) increasing -> a matching value is not found. (if *selectDiamMode* = *DIAM_EXT*)
16#0201_4018	The table of entered values is not monotonically (steadily) increasing -> 2 values are identical. (if *selectDiamMode* = *DIAM_EXT*)

Errors

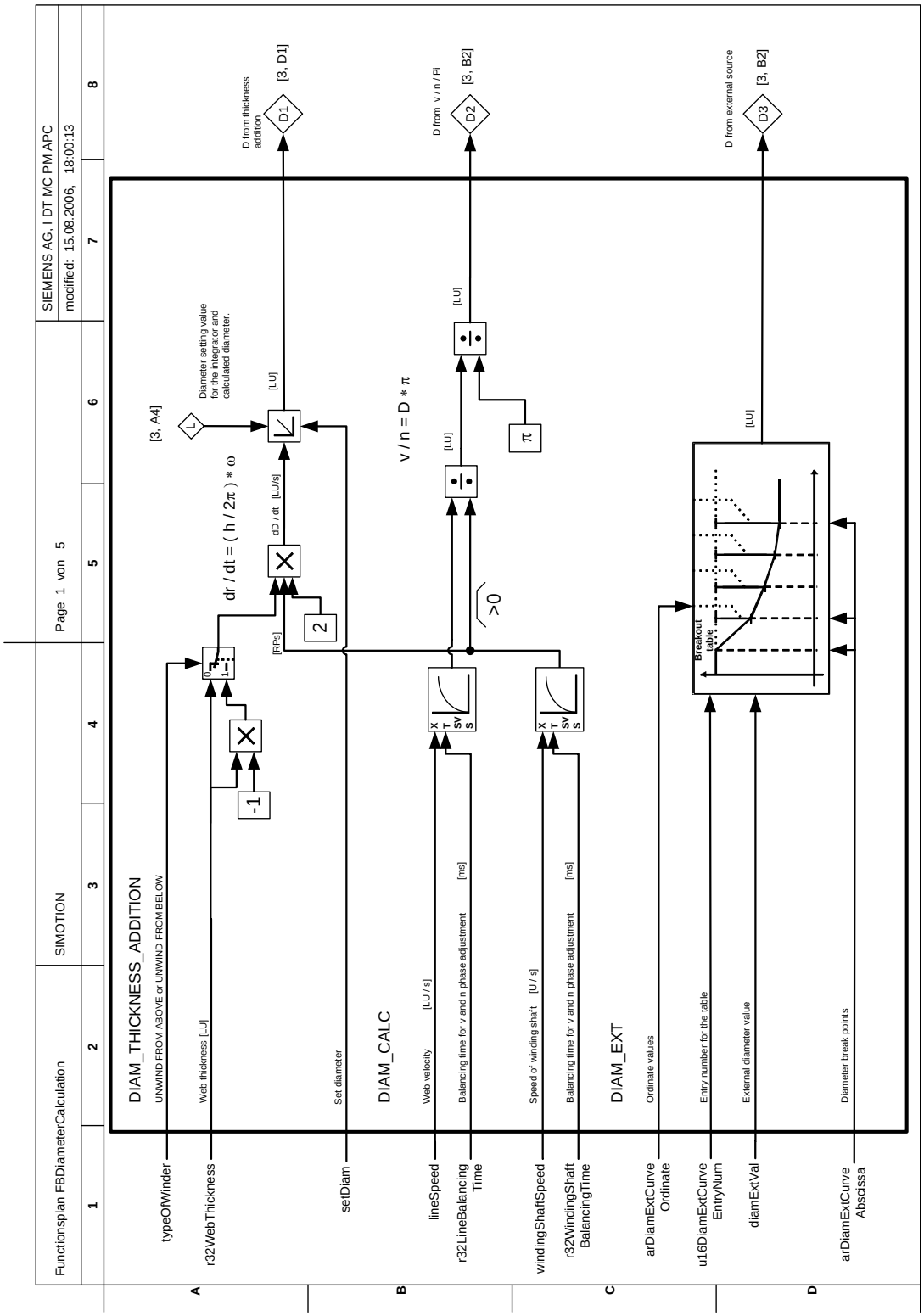
Error messages are indicated by the status of the outputs *error* and *errorID*: *error* = TRUE and *errorID* <> 16#0000_0000.

ErrorID	Description
16#0000_0000	No error
16#0201_8001	The function block is called in an incorrect task.
16#0201_8005	One or several configuration data are either not plausible or incorrect. *r32DiamMin* <= 0.0 *r32DiamMax* <= r32DiamMin *r32WebThickness* <= 0.0 *r32AdjustWebThickness* <= 0.0
16#0201_8007	Dimension units are incorrect/or are not defined DIAM_CALC_POSITION *sDiamConfig.r64WindingShaftFeedbackScal* <= 0.0 *sDiamConfig.r64LineFeedbackScal* <= *0.0*
16#0201_8013	The specified number of grid points is too low. (if *selectDiamMode* = *DIAM_EXT*)
16#0201_8014	The specified number of grid points is too high. (if *selectDiamMode* = *DIAM_EXT*)

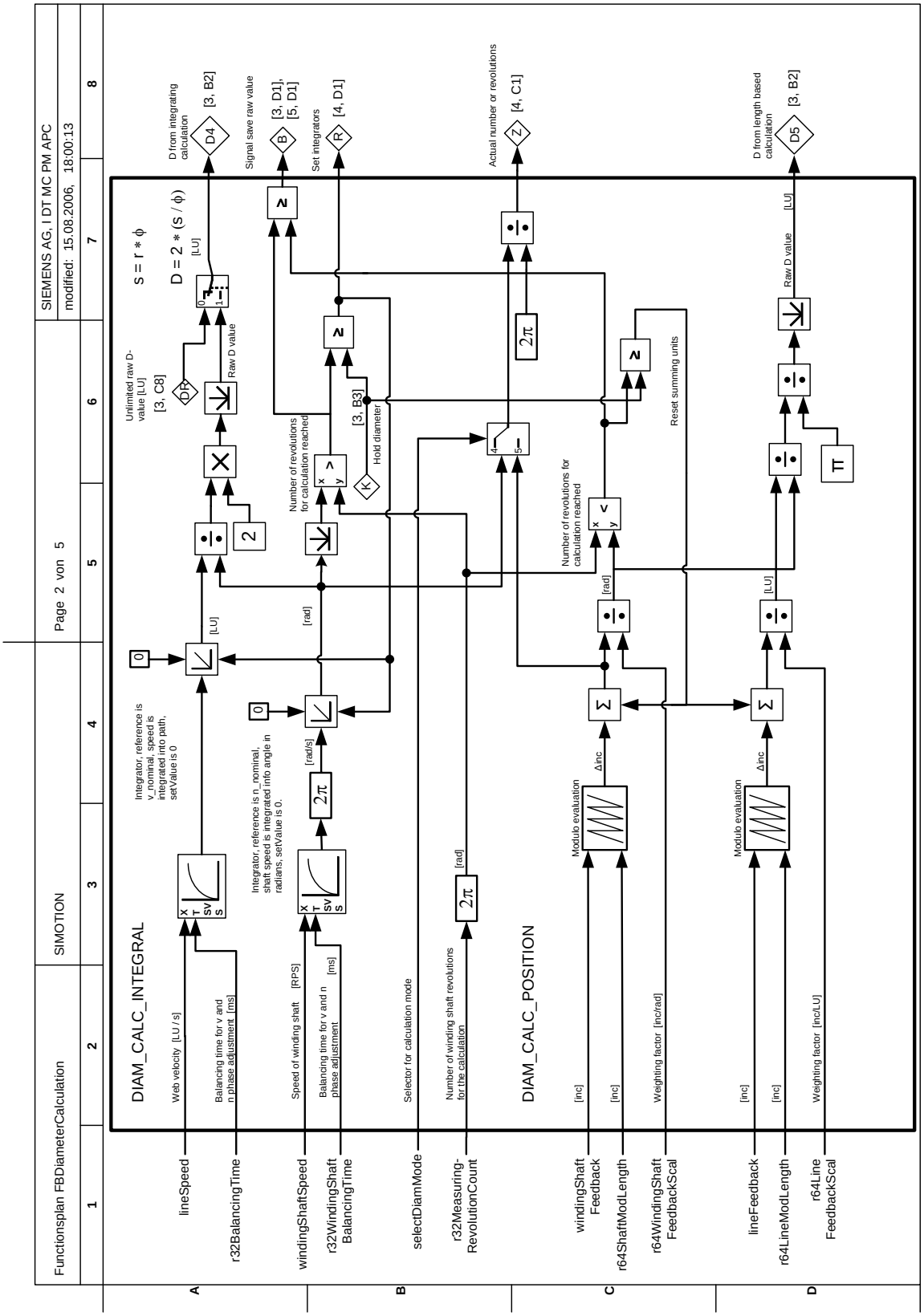
Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

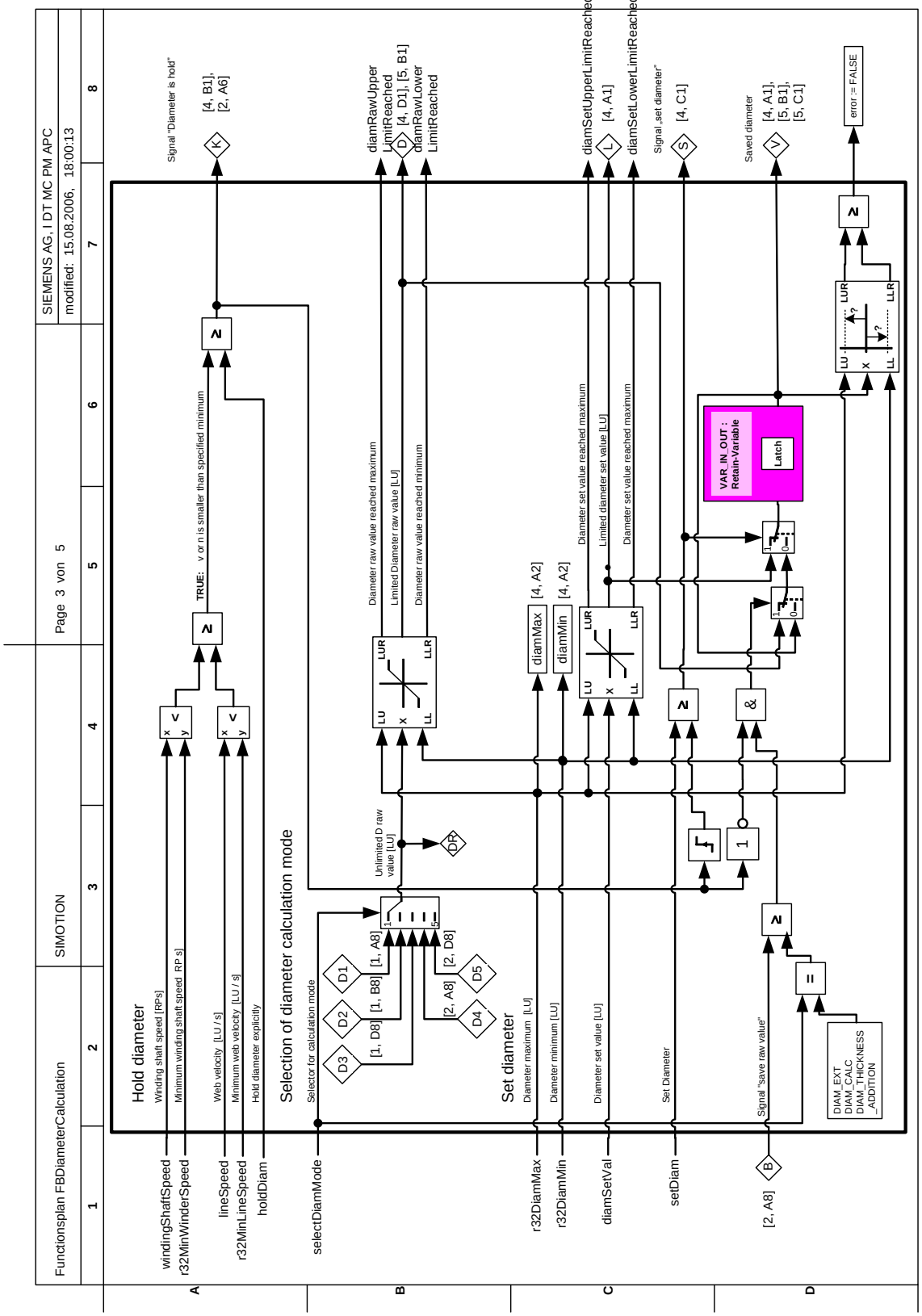
Function diagrams



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

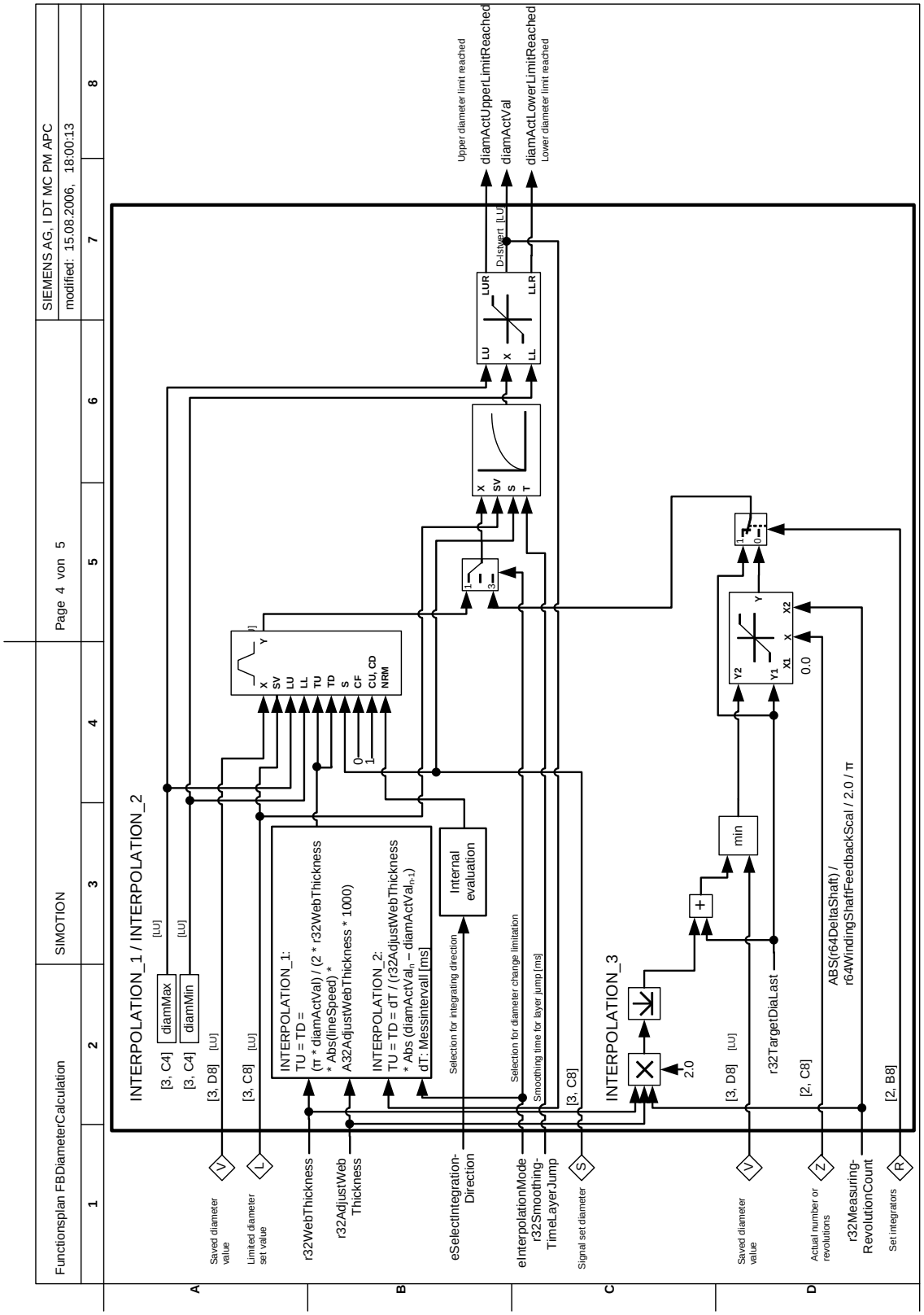

Functionsplan FBDiameterCalculation
SIMOTION
Page 2 von 5
SIEMENS AG, I DT MC PM APC
modified: 15.08.2006, 18:00:13
DIAM_CALC_INTEGRAL
DIAM_CALC_POSITION
lineSpeed
r32BalancingTime
windingShaftSpeed
r32WindingShaft BalancingTime
selectDiamMode
r32Measuring-RevolutionCount
windingShaft Feedback
r64ShaftModLength
r64WindingShaft FeedbackScal
lineFeedback
r64LineModLength
r64Line FeedbackScal
Web velocity [LU / s]
Balancing time for v and n phase adjustment [ms]
Speed of winding shaft [RPS]
Balancing time for v and n phase adjustment [ms]
Selector for calculation mode
Number of winding shaft revolutions for the calculation
Integrator, reference is v_nominal, speed is integrated into path, setValue is 0
Integrator, reference is n_nominal, shaft speed is integrated into angle in radians, setValue is 0.
Modulo evaluation
Reset summing units
Number of revolutions for calculation reached
Hold diameter
Unlimited raw D-value [LU] [3, C8]
Raw D value
D from integrating calculation [3, B2]
Signal save raw value [3, D1], [5, D1]
Set integrators [4, D1]
Actual number or revolutions [4, C1]
D from length based calculation [3, B2]
s = r * ϕ
D = 2 * (S / ϕ)

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc



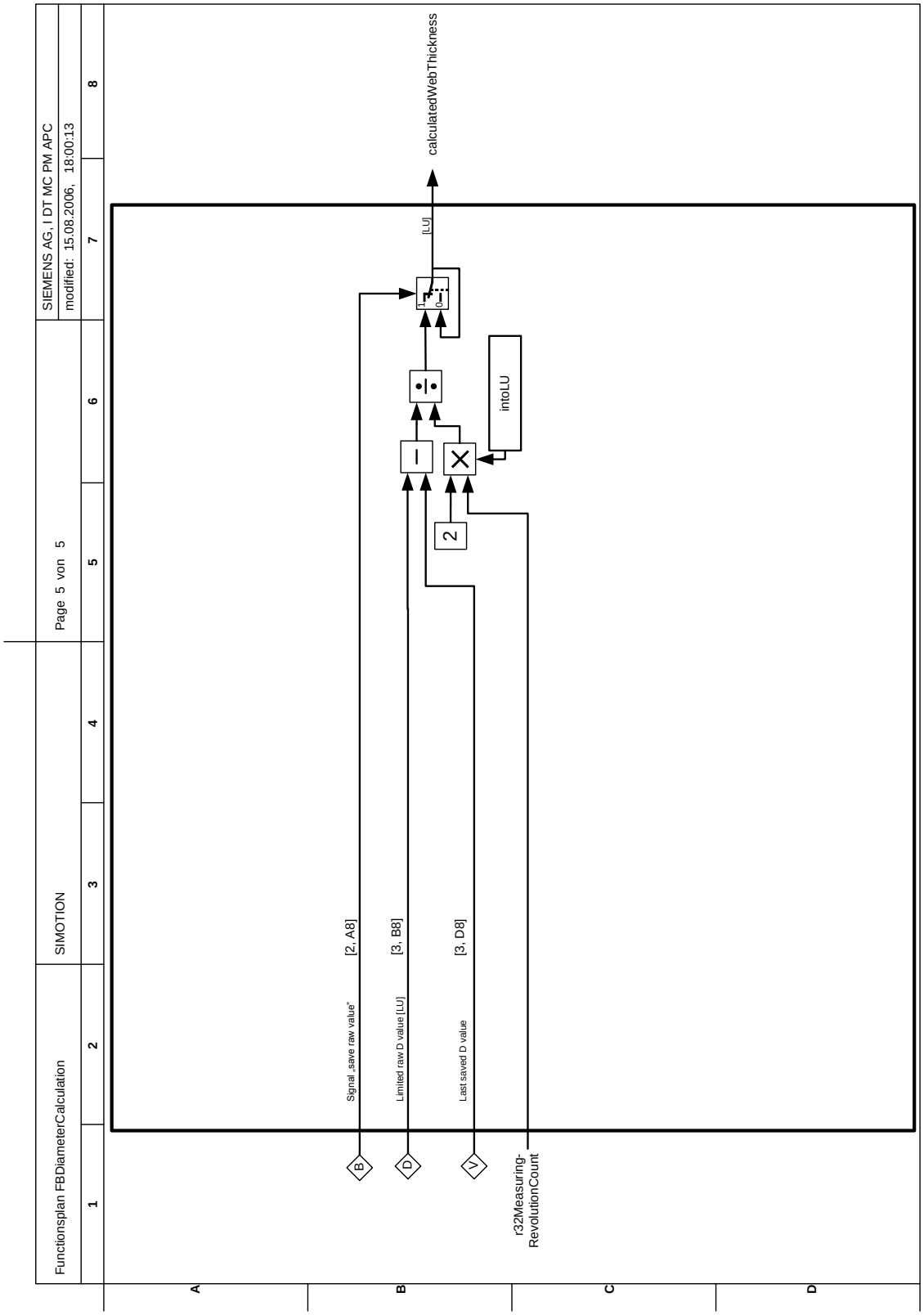
Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

5.2 FBTorquePrecontrol – Torque pre-control

Block name

FBTorquePrecontrol

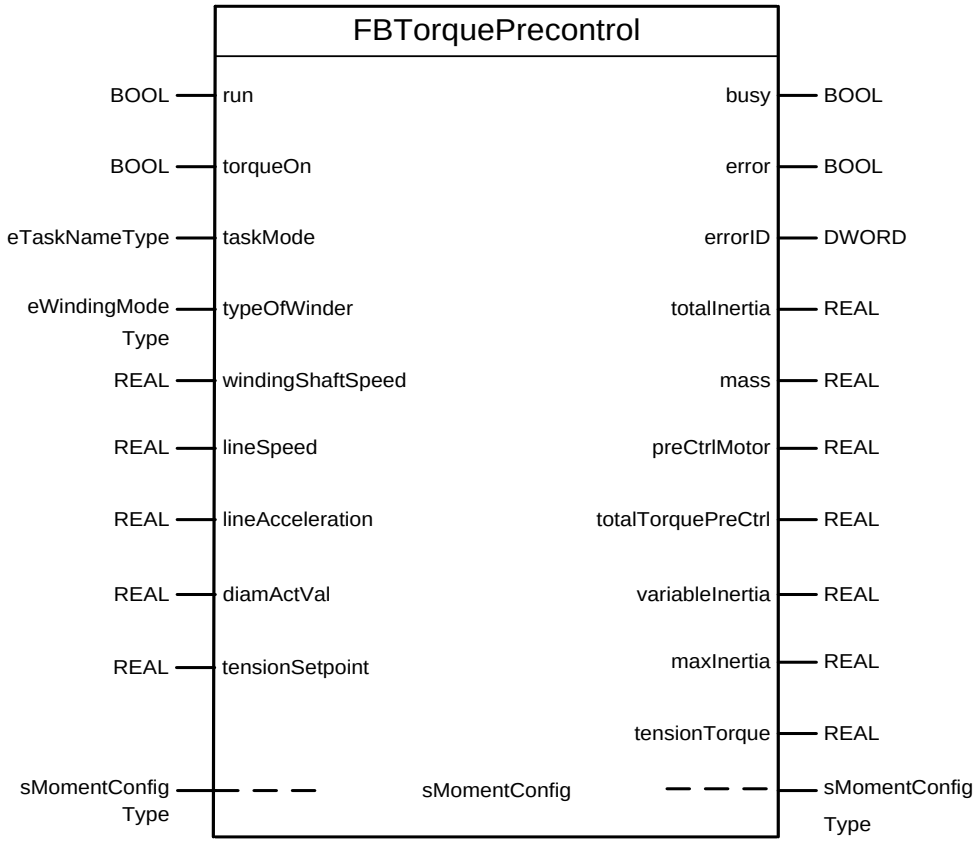
Task

This block calculates the variable and total moment of inertia, the mass of the roll being wound and the torque compensation to compensate for the inertia.

Integration in the run-time system

The function block must be called in a synchronous task (Servo, IPO, IPO_2 or Timer).

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Block parameters

Input parameters

Name	Data type	Initial value	Description
run	BOOL	FALSE	FALSE: Deactivates the block TRUE: Activates the block If the block has been de-activated, the calculated pre-control and tension torques are set to 0.
torqueOn	BOOL	FALSE	FALSE: Only the moment of inertia calculation is active TRUE: Moment of inertia and torque calculation are active Functions are only active, if run = TRUE!
taskMode	eTask NameType	IPO_ SYNCHRONO US_TASK	Task in which the FB is called: SERVO_SYNCHRONOUS_TASK: Servo Synchronous Task IPO_SYNCHRONOUS_TASK: Ipo Synchronous Task IPO_SYNCHRONOUS_TASK_2: Ipo Synchronous Task 2 TIMER_INTERRUPT_TASK: in a Timer Interrupt Task
typeOfWinder	eWinding ModeType	REWIND_ FROM_ ABOVE	Winder type: UNWIND_FROM_ABOVE: Unwinder from the top UNWIND_FROM_BELOW: Unwinder from the bottom REWIND_FROM_ABOVE: Winder from the top REWIND_FROM_BELOW: Winder from the bottom
windingShaftSpeed	REAL	0.0	[r / TU] / [° / TU] Winder shaft speed (for friction characteristic)
lineSpeed	REAL	0.0	[LUb / s] Web speed
lineAcceleration	REAL	0.0	[LUb / s²] Web acceleration
diamActVal	REAL	0.0	[LUb] Actual diameter
tensionSetpoint	REAL	0.0	[Z] Tension setpoint

Input-/Output Parameters

Name	Data type	Initial value	Description
sMomentConfig	sMoment ConfigType	-	Configuration data of the torque pre-control calculation Details on this structure are provided in the description of the type definitions.

Output parameters

Name	Data type	Initial value	Description
busy	BOOL	FALSE	FALSE: Function block deactivated TRUE: Function block active
error	BOOL	FALSE	FALSE: No error TRUE: An error has occurred, refer to error identification
errorID	DWORD	16#0000_0000	Error identification, refer to error messages errorID <= 16#000_8000: warning errorID >= 16#0000_8000: error
totalInertia	REAL	0.0	[mLUb²] Total moment of inertia referred to the motor shaft
mass	REAL	0.0	[m] Mass of the web
preCtrlMotor	REAL	0.0	[M] Torque precontrol

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Name	Data type	Initial value	Description
totalTorquePreCtrl	REAL	0.0	[M] Total precontrol torque based on motor shaft / Torque limitation
variableInertia	REAL	0.0	[mLUb²] Variable moment of inertia referred to the motor shaft
maxInertia	REAL	0.0	[mLUb²] Max. moment of inertia at max. Diameter referred to the motor shaft
tensionTorque	REAL	0.0	[M] Tension torque

Note

In principle, the calculations made in this function block are independent of any physical units. The input parameter determines the units of the output parameters. The internal calculations do not carry-out any conversions.

[LU]	Length units, e.g. m
[LUb]	Basis length unit, m or ft
[TU]	Time unit of the line-/winderAxis
[Z]	Tension, force, e.g. N
[m]	Mass, weight, e.g. kg, is defined by the units of density (specific gravity), e.g. kg/m³
[J] = mLUb²	Moment of inertia, e.g. kgm²
[M] = mLUb²s⁻²	Torque, e.g. Nm = kgm²s⁻²

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Functionality

The block determines the following:

- The variable moment of inertia for Kp adaptation of the speed and tension controller
- The pre-controlled torque for Δv/Δt and ΔD/Δt

The torque pre-control outputs the following torque setpoints to pre-control the winder drive:

- A torque setpoint as a function of the acceleration setpoint of the machine/system ramp-function generator and the instantaneous diameter actual value

- A torque setpoint as a function of Δn/Δt of the winder motor
  Note: This pre-control is especially required, if, e.g. thick material is to be wound onto a thin sleeve (i.e. mandrel) after a flying roll change. As a result of the diameter of the wound material that quickly increases, for the same consistent machine velocity, ΔD/Δt values of 20 x 1/s² can occur. These must be compensated. This also applies to unwinders.

Friction compensation:
The torque setpoint is generated depending on the shaft speed of the winder.

In order to quickly compensate for disturbances and to implement the closed-loop control in the first place (for a replacing speed control), the torque to be provided from the drive must be specified as accurately as possible. Depending on the control type, this is either realized additively after the speed controller or by dynamically adjusting the torque limits.

In both cases, starting from the weight of the wound material, the appropriate torque must be calculated using its moment of inertia (wound material) and transferred to the drive.

As an alternative to the method integrated in this block, the integrated friction characteristic of the drive system (SINAMICS) can also be used.

Note

The precise setting of the compensation is especially important for indirect tension control. This is so that the torque-generating current emulates, as far as possible, the material tension.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Calculations:

The function block *FBTorquePrecontrol* calculates the following physical quantities:

Weight of the wound material:

$$r32Mass = \frac{\pi \cdot r32Density \cdot r32Width \cdot \left(\text{diamActVal}^2 - r32\text{DiamMin}^2\right)}{4}$$

The material density can either be pre-assigned by selecting the material (selected using *sMomentConfig.eMaterialType* <> USER_DEFINED) or specified as value (*sMomentConfig.eMaterialType* = USER_DEFINED and with *sMomentConfig.r32Density* <> 0.0).

Moment of inertia of the wound material relating to the motor shaft:

$$\text{r32VariableInertia} = r32\text{Mass} \cdot \frac{\text{diamActVal}^2 + r32\text{DiamMin}^2}{8 \cdot r32GearRatio^2}$$

Total moment of inertia with variable moment of inertia for torque precontrol relating to the motor shaft:

$$\text{r32TotalInertia} = \frac{\text{r32JMandrel}}{\text{r32GearRatio}^2} + r32\text{VariableInertia} + r32\text{JMot} + r32\text{JFix}$$

Converting Library

Maximum moment of inertia at max diameter relating to the motor shaft:

$$mass_{\max} = \frac{\pi \cdot r32Density \cdot r32Width \cdot \left(\text{diamMax}^2 - r32\text{DiamMin}^2\right)}{4}$$

$$\text{variableInertia}_{\max} = \text{mass}_{\max} \cdot \frac{\text{r32DiamMax}^2 + r32\text{DiamMin}^2}{8}$$

$$\text{r32M}axInertia = \frac{\text{variableInertiamax} + r32\text{JMandrel}}{\text{r32GearRatio}^2} + r32\text{JMot} + r32\text{JFix}$$

Angular acceleration of the roll:

$$\alpha = 2 \cdot \frac{\text{lineAcceleration} \cdot r32\text{GearRatio}}{diamActVal}$$

Pre-control torque 1

$$M_{pre1} = r32TotalInertia \cdot \alpha$$

Pre-control torque 2 - Δd/Δt

$$M_{pre2} = \frac{2 \cdot r32TotalInertia \cdot r32GearRatio \cdot lineSpeed}{\frac{Tcyc}{1000}} \cdot \left(\frac{1}{diamActVal_n} - \frac{1}{diamActVal_{n-1}}\right)$$

$$M_{pre2} = M_{pre2} * r32DndtAdjust$$

Speed-dependent frictional torque of the winder:

$$M_{frict} = M_{frict}(n)$$

This frictional torque is internally determined using the function block *FBPolygonA*. The user is responsible for providing valid and sensible values in the appropriate data fields. If the factor s*MomentConfigStruc.r32FrictionAdjust* is set to 0.0, *FBPolygonA* is not called and therefore a frictional torque is not added.

Finally, depending on the winder type, the pre-control torque of the motor, the sum of all pre-control torques and the tension torque are calculated.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

The winding mode is always defined assuming a positive line speed. If the leading value becomes negative the winding mode will be inverted automatically (a rewinder becomes an unwinder and vice versa). The same applies for winding *from above* and *from below*. This inversion is automatic.

Rewinding from above

$$M_{preCtrlMotor} = M_{pre1} - M_{pre2} + M_{frict}$$

$$M_{tensionTorque} = M_{tensionTorque}$$

$$M_{totalTorquePreCtrl} = M_{preCtrlMotor} + M_{tensionTorque}$$

Rewinding from below

$$M_{preCtrlMotor} = -M_{pre1} + M_{pre2} - M_{frict}$$

$$M_{tensionTorque} = -M_{tensionTorque}$$

$$M_{totalTorquePreCtrl} = M_{preCtrlMotor} + M_{tensionTorque}$$

Unwinding from above

$$M_{preCtrlMotor} = M_{pre1} + M_{pre2} + M_{frict}$$

$$M_{tensionTorque} = -M_{tensionTorque}$$

$$M_{totalTorquePreCtrl} = M_{preCtrlMotor} + M_{tensionTorque}$$

Unwinding from below

$$M_{preCtrlMotor} = -M_{pre1} - M_{pre2} - M_{frict}$$

$$M_{tensionTorque} = M_{tensionTorque}$$

$$M_{totalTorquePreCtrl} = M_{preCtrlMotor} + M_{tensionTorque}$$

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Warnings

Warnings are indicated by the status of the outputs *error* and *errorID*: *error* = FALSE and *errorID* <> 16#0000_0000.

ErrorID	Description
16#0000_0000	No warning
16#0203_4015	The table of entered values is not monotonically (steadily) increasing -> no matching value found. (friction characteristic)
16#0203_4018	The table of entered values is not monotonically (steadily) increasing -> 2 values are identical. (friction characteristic)

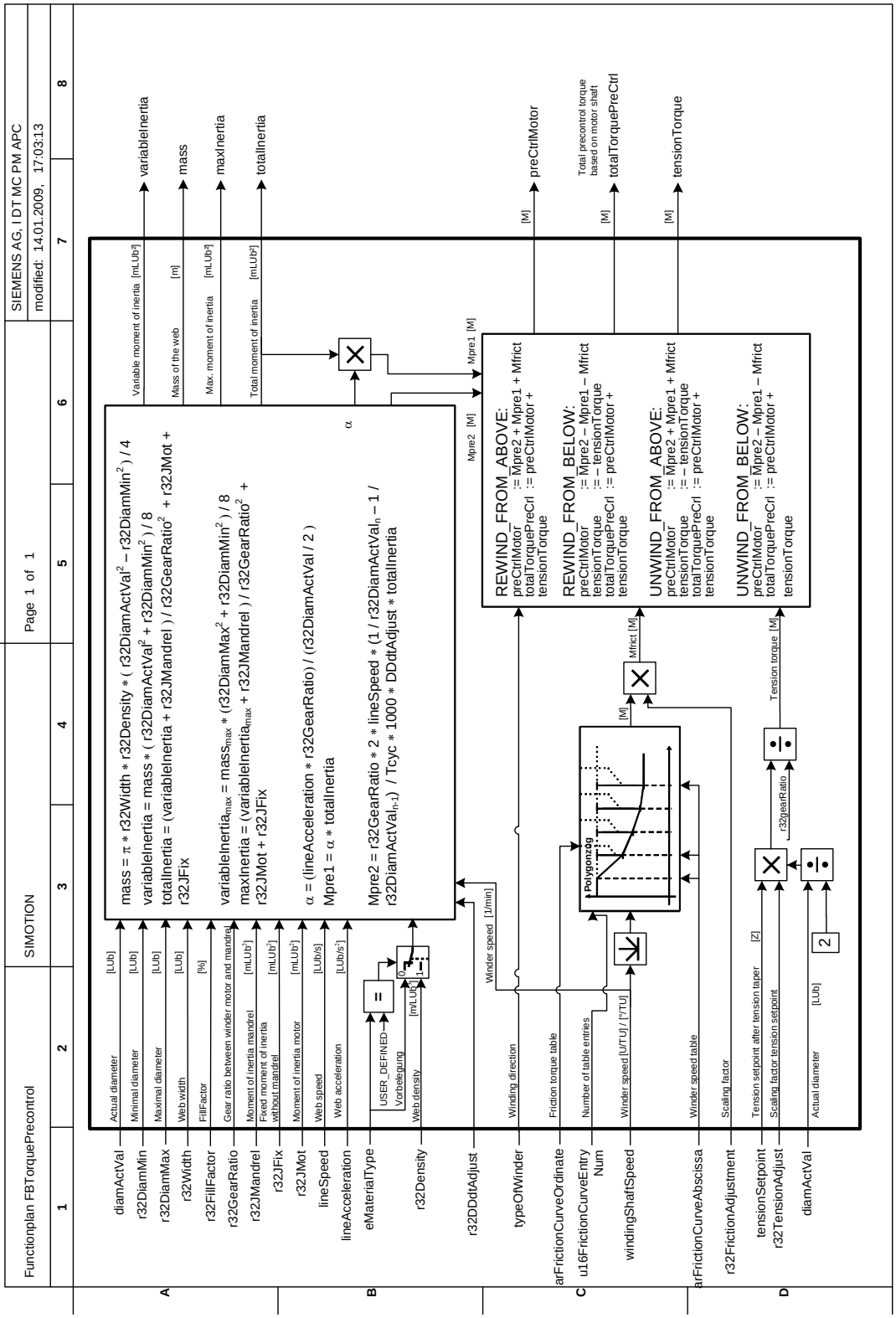
Errors

Error messages are indicated by the status of the outputs *error* and *errorID*: *error* = TRUE and *errorID* <> 16#0000_0000.

ErrorID	Description
16#0000_0000	No error
16#0203_8001	The function block is called in an incorrect task.
16#0203_8005	One or several configuration data are not plausible or lie outside the specified limits: *sMomentConfig.r32GearRatio* <= 0.0 *sMomentConfig.r32Density* < 0.0 *sMomentConfig.r32DiamMin* <= 0.0 *sMomentConfig.r32DiamMax* < *sMomentConfig.r32DiamMax* *sMomentConfig.r32Width* < 0.0 *sMomentConfig.r32fillFactor* <= 0.0 *sMomentConfig.r32JFix* < 0.0 *sMomentConfig.r32JMandrel* < 0.0 *sMomentConfig.r32JMot* < 0.0
16#0203_8006	The diameter actual value is not plausible *sMomentConfig.r32DiamActVal* <= 0.0 *sMomentConfig.r32DiamActVal* < *sMomentConfig.r32DiamMin*
16#0203_8013	The number of grid points that have been entered is too small. (friction characteristic)
16#0203_8014	The number of grid points that have been entered is too high. (friction characteristic)

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Function diagram



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

5.3 FBTaperCharacteristic – Tension taper characteristic

Block name

FBTaperCharacteristic

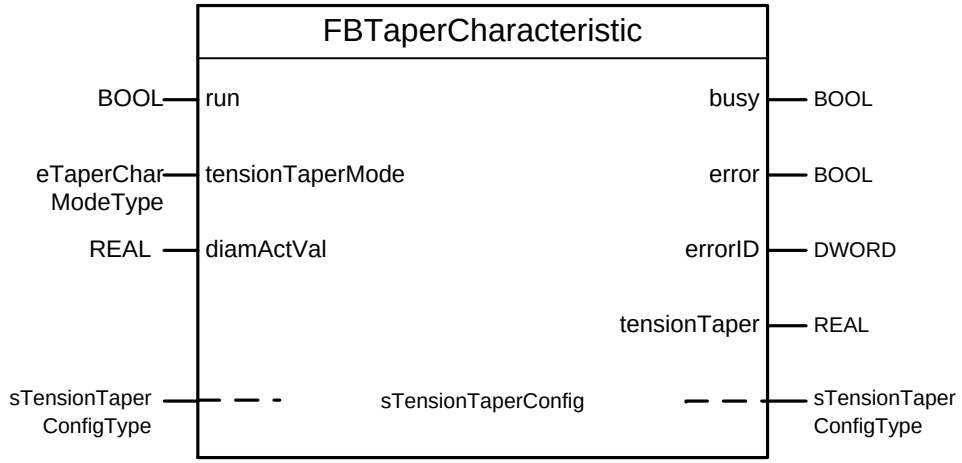
Task

This function block calculates a reduction factor for the tension setpoint as a function of the diameter.

Integration in the run-time system

The function block can be called in any task.

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Block parameters

Input parameters

Name	Data type	Initial value	Description
run	BOOL	FALSE	FALSE: The block is not processed TRUE: The block is processed If the block is de-activated (run = FALSE) the output tensionTaper is set to 1.0.
tensionTaper Mode	eTaperChar ModeType	TAP_ CHAR_ NO_ TAPER	Tension taper characteristic mode: TAP_CHAR_NO_TAPER: No characteristic TAP_CHAR_LINEAR: Linear characteristic TAP_CHAR_HYPERBOLIC_1: Hyperbolic char.1 TAP_CHAR_HYPERBOLIC_2: Hyperbolic char.2 TAP_CHAR_TABLE : Characteristic defined by a table
diamActVal	REAL	0.0	[%] / [LU] Actual diameter value

Converting Library

Input-/Output Parameters

Name	Data type	Initial value	Description
sTensionTaper Config	sTension TaperConfig Type		Structure with configuration data of the tension taper characteristic Details on this structure are provided in the description of the type definitions.

Output parameters

Name	Data type	Initial value	Description
busy	BOOL	FALSE	FALSE: function block inactive The output tensionTaper is set to 1.0, actual errors are deleted. TRUE: function block active
error	BOOL	FALSE	FALSE: No error TRUE: An error has occurred, refer to error identification In case of error (error = TRUE) the output tensionTaper is set to 1.0.
errorID	DWORD	16#0000_0000	Error identification, refer to error messages errorID <= 16#000_8000: warning errorID >= 16#0000_8000: error
tensionTaper	REAL	1.0	[-] Taper adjustment factor

Note

In principle, the calculations made in this function block are independent of any physical units. The input parameter determines the units of the output parameters. The internal calculations do not carry-out any conversions.

[LU] Length units, e.g. m
[%] Percentage, e.g. 100%

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Functionality

The tension taper characteristic is used to define with which tension the material should be wound onto the roll as a function of the roll diameter. It only makes sense to use the tension taper characteristic while actually winding in order to wind the inner layers tightly.

For dancer roll controls, the tension taper characteristic can be used to reduce the pressure setpoint for the dancer roll support. In this case, the output of the characteristic must be multiplied with the tension setpoint and made available at the analog output.

The block is activated using the *run* input. Depending on the mode selected at input *tensionTaperMode*, the tension reduction factor is calculated as a function of the diameter present *diamActVal* and output in *tensionTaper*.

The way in which the reduction factor is processed depends on the mode that has been selected.

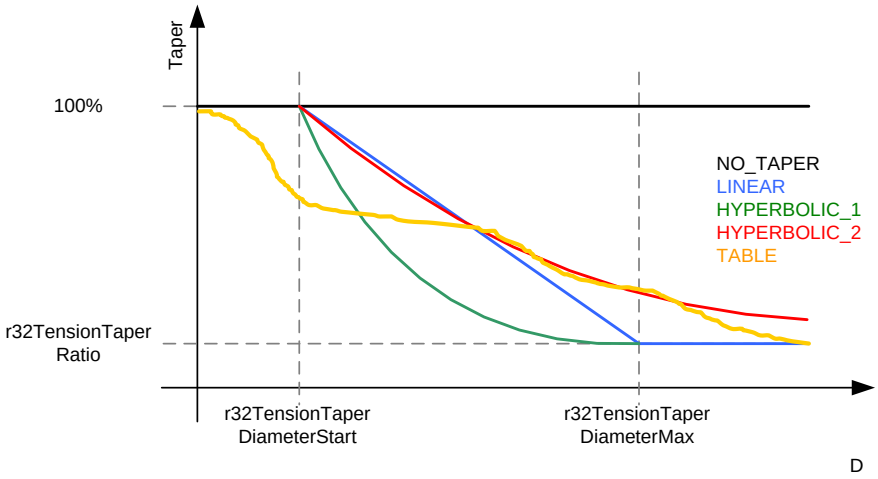
Converting Library

Table 5-10: Calculation types of the winding hardness characteristic (taper)

Calculation mode	Description
TAP_CHAR_NO_TAPER	1.0 is assigned for the reduction factor for the tension setpoint. $tensionTaper = 1.0$
TAP_CHAR_LINEAR	From a starting diameter that has to be entered *r32TensionTaperDiameterStart,* the reduction factor is linearly reduced to the maximum diameter *r32TensionTaperDiameterMax* with a reduction of *r32TensionTaperRatio*. *diamActVal* < *r32TensionTaperDiameterStart*: $tensionTaper = 1.0$ *r32TensionTaperDiameterStart* < *diamActVal* < *r32TensionTaperDiameterMax*: $tensionTaper = \cdot\left(1 - \frac{r32TensionTaperRatio}{100} \cdot \left(\frac{diamActVal - r32\text{TensionTaperDiameterStart}}{\text{r32TensionTaperDiameterMax - r32TensionTaperDiameterStart}}\right)\right)$ *diamActVal* > *r32TensionTaperDiameterMax*: $tensionTaper = (1 - r32\text{TensiontaperRatio} / 100.0)$
TAP_CHAR_HYPERBOLIC_1	From a starting diameter that has to be entered *r32TensionTaperDiameterStart* the reduction factor is reduced as hyperbolic function with *r32TensionTaperRatio* to the maximum diameter *r32TensionTaperDiameterMax*. In this case, the reduction is at a maximum when the maximum diameter has been reached. *diamActVal* < *r32TensionTaperDiameterStart*: $tensionTaper = 1.0$ *r32TensionTaperDiameterStart* < *diamActVal* < *r32TensionTaperDiameterMax*: $tensionTaper = \cdot\left(\left(1 - \frac{r32TensionTaperRatio}{100} \cdot \left(\frac{diamActVal - r32\text{TensionTaperDiameterStart}}{diamActVal}\right)\right) \cdot \left(\frac{r32TensionTaperDiameterMax}{r32TensionTaperDiameterMax - r32TensionTaperDiameterStart}\right)\right)$ *diamActVal* > *r32TensionTaperDiameterMax*: $tensionTaper = (1 - r32\text{Tensiontaperratio} / 100.0)$
TAP_CHAR_HYPERBOLIC_2	From a starting diameter that has to be entered *r32TensionTaperDiameterStart* the reduction factor is reduced as hyperbolic function with *r32TensionTaperRatio* to the maximum diameter *r32TensionTaperDiameterMax*. In this case, the full reduction is never reached. The characteristic is only approached. *diamActVal* < *r32TensionTaperDiameterStart*: $tensionTaper = 1.0$ *diamActVal* >*r32TensionTaperDiameterStart* $tensionTaper = \left(1 - \frac{r32TensionTaperRatio}{100} \cdot \left(1 - \frac{r32\text{TensionTaperDiameterStart}}{diamActVal}\right)\right)$
TAP_CHAR_TABLE	The reduction factor is determined by entering a characteristic. In this characteristic, a diameter is assigned a tension reduction. The characteristic is saved in the data structures *arTensionTaperCurveAbscissa* and *arTensionTaperCurveOrdinate*. The characteristic is linearly interpolated between the two points on the characteristic.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Figure 5-1: Characteristic types:



Warnings

Warnings are indicated by the status of the outputs *error* and *errorID*: *error* = FALSE and *errorID* <> 16#0000_0000.

ErrorID	Description
16#0000_0000	No warning
16#0205_4015	The table of entered values is not monotonically (steadily) increasing -> no matching value found. (if *tensionTaperMode* = *TAP_CHAR_TABLE*)
16#0205_4018	The table of entered values is not monotonically (steadily) increasing -> 2 values are identical. (if *tensionTaperMode* = *TAP_CHAR_TABLE*)

Errors

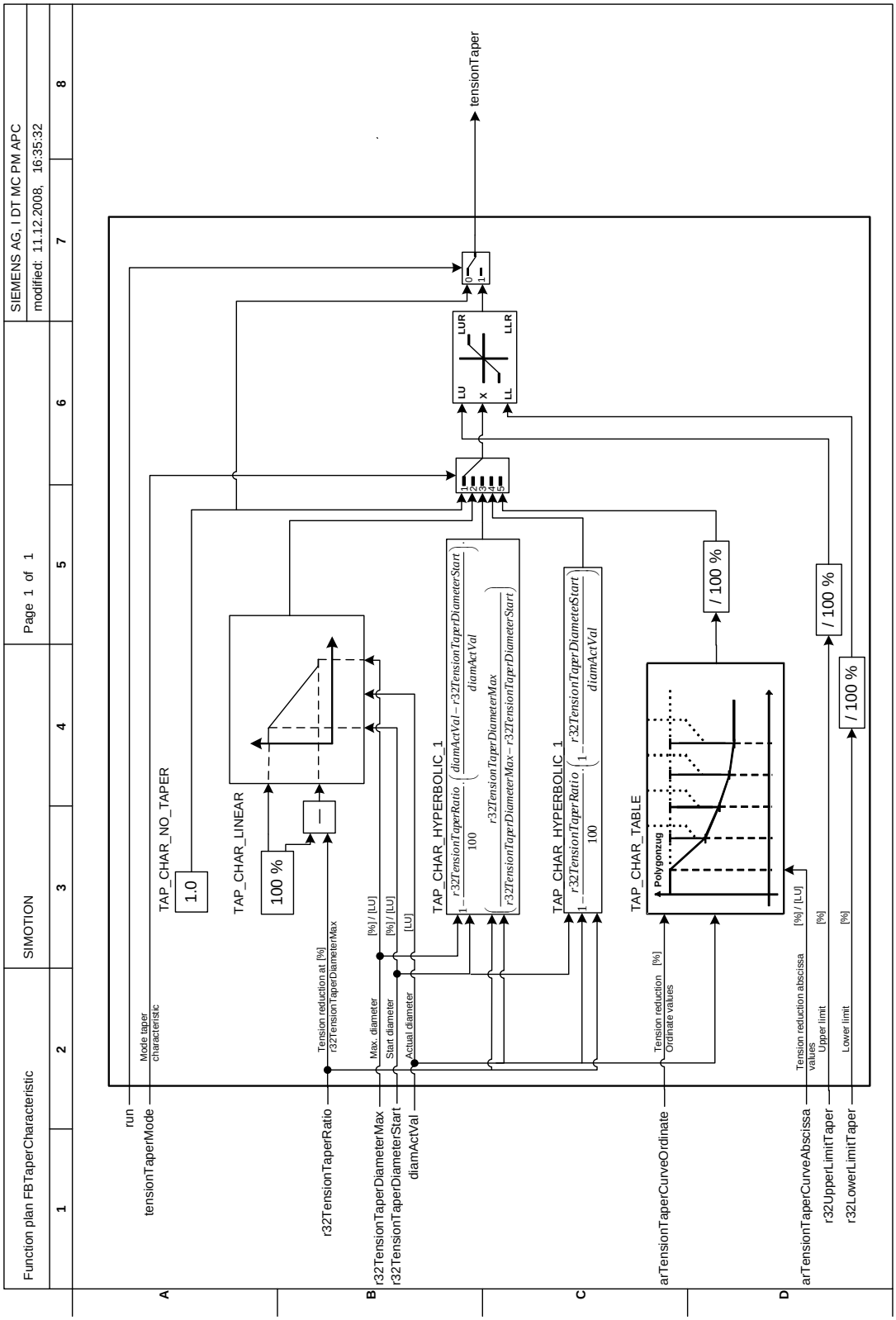
Error messages are indicated by the status of the outputs *error* and *errorID*: *error* = TRUE and *errorID* <> 16#0000_0000.

ErrorID	Description
16#0000_0000	No error
16#0205_8005	Invalid input parameter. *diamActVal* <= 0.0 *sTensionTaperConfig.r32UpperLimitTaper* < *sTensionTaperConfig.r32LowerLimitTaper* *sTensionTaperConfig.r32LowerLimitTaper* < 0.0 For TAP_CHAR_LINEAR, TAP_CHAR_HYPERBOLIC_1, TAP_CHAR_HYPERBOLIC_2: *sTensionTaperConfig.r32TensionTaperDiameterStart* < 0.0 *sTensionTaperConfig.r32TensionTaperDiameterMax* <= *sTensionTaperConfig.r32TensionTaperDiameterStart* *sTensionTaperConfig.r32TensionTaperRatio* < 0.0 *sTensionTaperConfig.r32TensionTaperRatio* > 100.0
16#0205_8013	The number of grid points that has been entered is too small. (if *tensionTaperMode* = *TAP_CHAR_TABLE*)
16#0205_8014	The number of grid points that has been entered is too high. (if *tensionTaperMode* = *TAP_CHAR_TABLE*)

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Function diagram



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

5.4 FCRLC – Residual length calculation

The residual length of a roll can be calculated using this block.

Block name

FCRLC

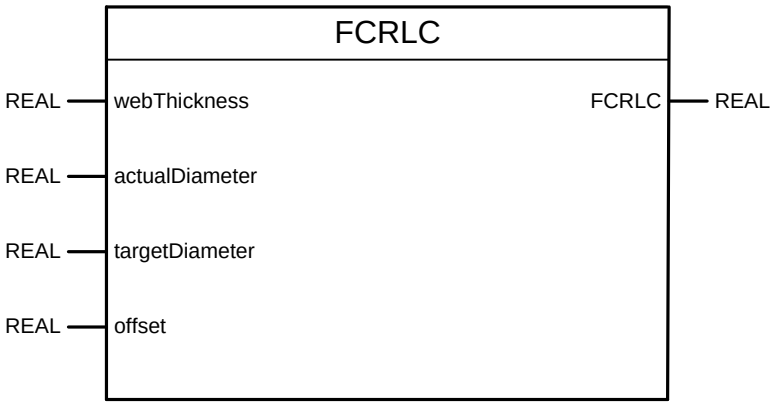
Task

The length of residual material on a roll can be calculated by entering the appropriate roll parameters.

Integration in the run-time system

The function can be called in any task.

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Block parameters

Input parameters

Name	Data type	Initial value	Description
webThickness	REAL	0.0001	[LU] Thickness of the material on the roll [0.000001–1]
actualDiameter	REAL	-	[LU] Actual diameter of the roll [-]
targetDiameter	REAL	-	[LU] Target diameter to be reached [-]
offset	REAL	0.0	[LU] Offset to the calculated residual length [-]

Output parameters

Name	Data type	Initial value	Description
FCRLC	REAL	-	[LU] Calculated residual length of the roll [-]

Note

The calculations made in this function block are in principle independent of the physical units. The output parameter units are defined by the input parameters. The internal calculations do not perform any conversions.

[LU] Length units, e.g. m

Functionality

Depending on the roll parameters that have been entered, the function calculates the residual length (*FCRLC*). To do this, the following data is required – the thickness of the material on the roll (*webThickness*), the actual diameter (*actualDiameter*) and the diameter to be reached (*targetDiameter*). In this case, the target diameter represents the end of the roll or the diameter to be reached. Depending on whether the target diameter is greater than or less than the actual diameter, the length of the material on the roll or the length of the material still to be wound is output. The thickness of the material web (*webThickness*) is limited using the internal constants *MIN_THICKNESS* and *MAX_THICKNESS*.

$$FCRLC = \frac{\Pi}{4 * webThickness} * \left(actualDiameter^2 - t\arg etDiameter^2\right) + offset$$

Figure 5-2: Graphic representation of the function

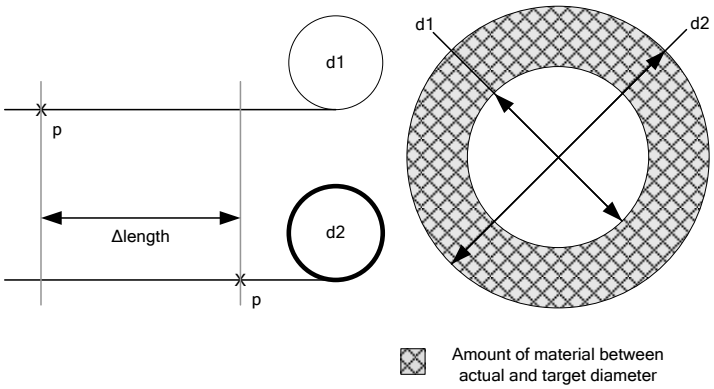


Table 5-11

Winder mode	Actual diameter	Target diameter
Winder	d1 < *d2*	d2 (full roll)
Unwinder	d2 > *d1*	d1 (empty roll)

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

5.5 FBCalcTargetDiamStopCondition

Block name

FBCalcTargetDiamStopCondition

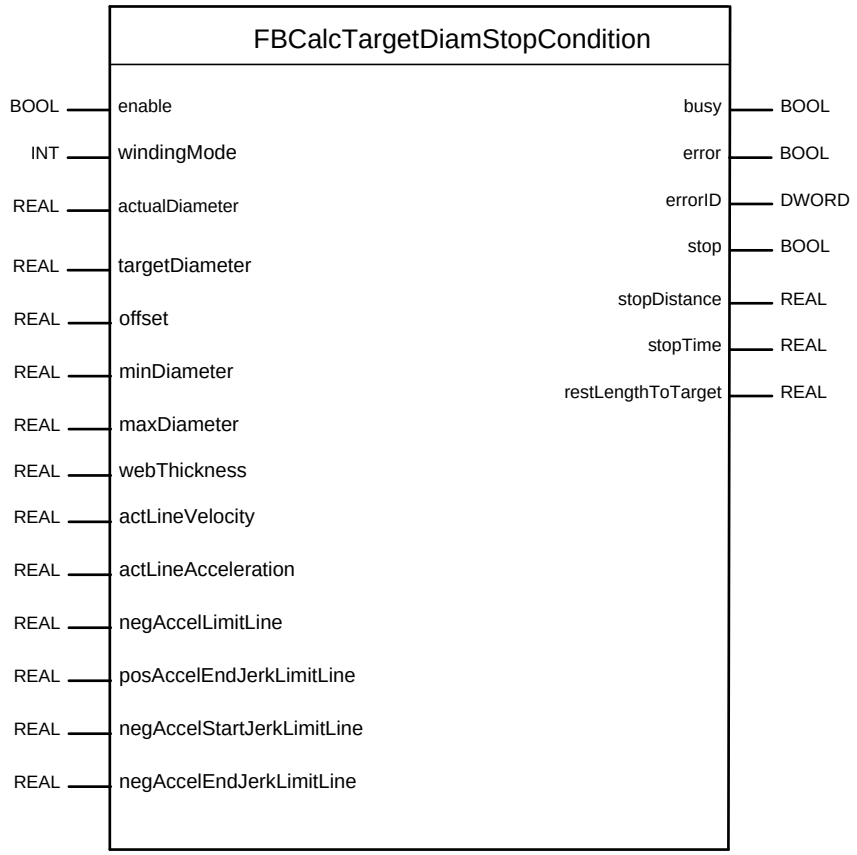
Task

The function block calculates the rest length to a specified target diameter and the necessary stopping time.

Integration in the run-time system

The function can be called in any task.

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Block parameters

Input parameters

Name	Data type	Initial value	Description
enable	BOOL	FALSE	TRUE = enable Function Block, False = Disable
windingMode	INT	0	0: rewind, 1: unwind hint: distinguish dependent on line speed sign
actualDiameter	REAL	0.0	[LU] actual diameter value
targetDiameter	REAL	0.0	[LU] target diameter value
offset	REAL	0.0	[LU] offset rest length to target diameter
minDiameter	REAL	0.0	[LU] Minimum diameter allowed
maxDiameter	REAL	0.0	[LU] Maximum diameter allowed
webThickness	REAL	0.0	[LU] Web thickness for rest length calculation
actLineVelocity	REAL	0.0	[LU / min] actual velocity (of line) has to be positive hint: (use ABS())
actLineAcceleration	REAL	0.0	[LU / min²] actual acceleration (of line)
negAccelLimitLine	REAL	0.0	[LU / min²] Deceleration limit (of line), has to be negative
posAccelEndJerkLimitLine	REAL	0.0	[LU / min^3] Jerk limit (of line), has to be negative or zero
negAccelStartJerkLimitLine	REAL	0.0	[LU / min^3] Jerk limit (of line), has to be negative or zero
negAccelEndJerkLimitLine	REAL	0.0	[LU / min^3] Jerk limit (of line), has to be positive or zero

Output parameters

Name	Data type	Initial value	Description
busy	BOOL	-	TRUE = Function Block calculates values, FALSE = Funciton Block idle
error	BOOL	-	TRUE = error occured, FALSE = no error
errorID	DWORD	-	Error identification
stop	BOOL	-	TRUE = target reached, FALSE = target not reached
stopDistance	REAL	-	[LU] distance to stop line with actual velocity and dynamics
stopTime	REAL	-	[s] time to decelerate line to velocity 0
restLengthToTarget	REAL	-	[LU] rest length to target diameter

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Functionality

The function block combines internal the the LConLib functions FCRLC and FCCalculateDistance.

With a rising edge at the input *enable* a plausibility check of the input parameter will be started. Faults will be displayed at the outputs *error* and *errorID*.

Rest length calculation:

The rest length calculation is parameterized with the inputs *webThickness*, *actualDiameter*, *targetDiameter* and *offset*. The specified target diameter is limited with *minDiameter* and *maxDiameter*. The calculated rest length is output at the FB output *restLengthToTarget*.

NOTE The rest length calculation is continued even there is fault active (error= TRUE).

Stopping distance calculation:

The stopping distance *(stopDistance)* and stopping time *(stopTime)* is being calculated dependent on the actual line speed and the active dynamics.

As soon the calculated stopping distance is smaller than the calculated rest length the FB output *stop* comes TRUE.

NOTICE **The line axis need to be stopped by the user in case stop = TRUE!**

With a falling edge at *enable* all ouput parameters will be reset.

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

5.6 FCRDC – Target diameter calculation

Calculating the target diameter of a roll.

Block name

FCRDC

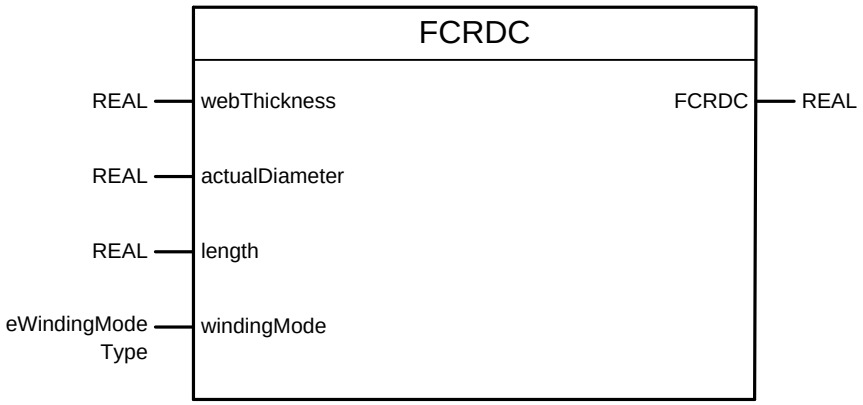
Task

With the input of the parameters the target diameter of a roll can be calculated.

Integration in the run-time system

The function can be called in any task.

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Block parameters

Input parameters

Name	Data type	Initial value	Description
webThickness	REAL	0.0001	[LU] Material thickness [0.000001–1]
actualDiameter	REAL	0.0	[LU] Actual roll diameter
length	REAL	0.0	[LU] target length to be wound/unwound
windingMode	eWindingMode Type	REWIND_ FROM_ ABOVE	Winding mode (Winder or unwinder)

Output parameters

Name	Data type	Initial value	Description
FCRDC	REAL	-	[LU] Calculated target diameter of the roll

Note

The calculations made in this function block are in principle independent of the physical units. The output parameter units are defined by the input parameters. The internal calculations do not perform any conversions.

[LU] Length units, e.g. m

Functionality

The function uses the input parameters to calculate the target diameter (*FCRDC*) of a roll. The calculation relies on the material thickness (*webThickness*), the actual roll diameter (*actualDiameter*) and the material length to be wound or unwound (*length*). The winding mode is defined using the *windingMode* input. The material thickness (*webThickness*) is internally limited (*MIN_THICKNESS, MAX_THICKNESS*).

In case of a too high material length input the function result is zero.

Winder

$$FCRDC = 2 * \sqrt{\frac{actualDiameter^2}{4} + \frac{webThickness * length}{\Pi}}$$

Unwinder

$$FCRDC = 2 * \sqrt{\frac{actualDiameter^2}{4} - \frac{webThickness * length}{\Pi}}$$

Figure 5-3: Graphic representation of the function

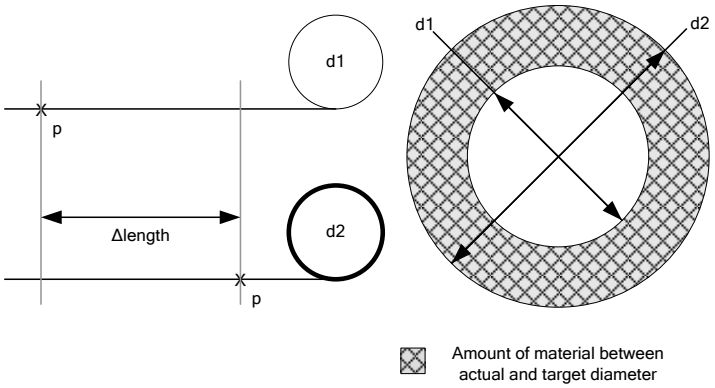


Table 5-12

Winder mode	Actual diameter	Target diameter
Winder	d1 < *d2*	d2 (full roll)
Unwinder	d2 > *d1*	d1 (empty roll)

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

5.7 FBThicknessAverage – High resolution thickness calculation

The function block calculates a floating average value based on the cyclically calculated material thickness values of the diameter calculator FBDiameterCalculation, to increase the precision and stability of the value.

Block name

FBThicknessAverage

Task

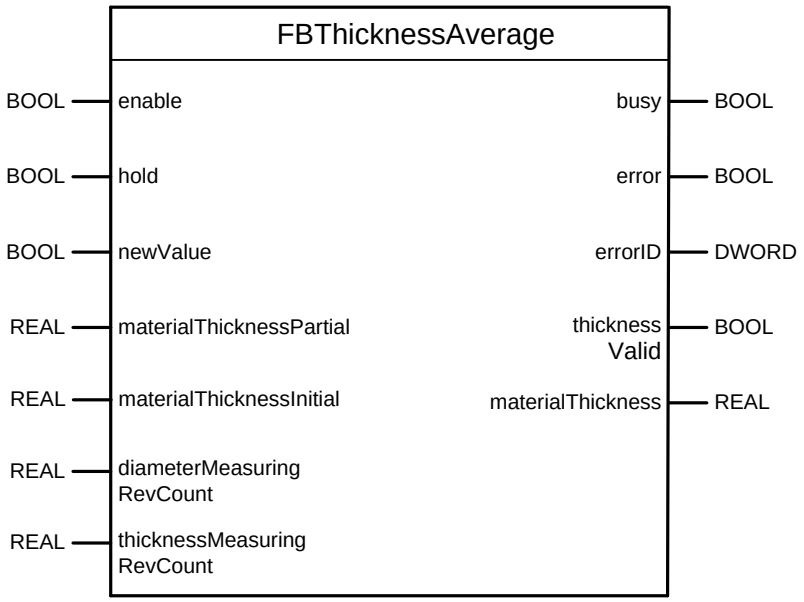
The block is used as extension for the FBWinder winder block or the FBDiameterCalculation diameter calculator.

Both of these blocks offer several calculation modes for calculating the diameter, whereby the DIAM_CALC_INTEGRAL and DIAM_CALC_POSITION position modes, not only calculate the diameter, but also the actual material thickness. The FBThicknessAverage block improves the precision of the material thickness calculation by generating a floating average value (extension of the measurement interval), in order to make the value more suitable - for instance to calculate the remaining length.

Integration in the run-time system

The function block can be called in any task.

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Block parameters

Input parameters

Name	Data type	Initial value	Description
enable	BOOL	FALSE	TRUE: Block is active. A rising edge resets the block and fills the elements of the ring buffer with *materialThicknessInitial*. FALSE: Block inactive, output *materialThickness* = *materialThicknessInitial*.
hold	BOOL	FALSE	TRUE: Block does not accept any new values, e.g. if the winder is stopped or the machine velocity is changed.
newValue	BOOL	FALSE	Rising edge: The value in *materialThicknessPartial* is written into the ring buffer.
materialThicknessPartial	REAL	0.0	Actual material thickness calculated by the diameter calculator
materialThicknessInitial	REAL	0.0	Initial value of the material thickness calculation
diameterMesuringRevCount	REAL	1.0	Number of winder revolutions of the diameter calculation
thicknessMeasuringRevCount	REAL	1.0	Number of winder revolutions of the material thickness calculation

Output parameters

Name	ata type	itial value	Description
		E	**FB is active**
		E	**E: No error** **An error has occurred, refer to error identification**
D	**RD**	**00_0000**	**dentification, refer to error messages** **D <= 16#000_8000: warning** **D >= 16#0000_8000: error**
essValid		E	**uffer is filled with measured values**
alThickness			**ated material thickness if FB is active, *materialThicknessInitial* if not.**

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Functionality

The following equation shows the block functionality:

$$mt_i = \frac{|\Delta d_i|}{2*\#d} \qquad k = \left\lfloor \frac{\#t}{\#d} \right\rfloor$$

$$MT = \frac{|\Delta D|}{2*\#t} = \sum_k \frac{mt_i}{k} = \sum_k \frac{|\Delta d_i|}{2*\#d*k} = \frac{|\Delta D|}{2*\#d*k} = \frac{|\Delta D|}{2*\#t}$$

$$\Delta D = \sum_k \Delta d_i$$

MT: Material thickness, calculated from FBThicknessAverage, updated after the specified number of winder revolutions (floating average value)

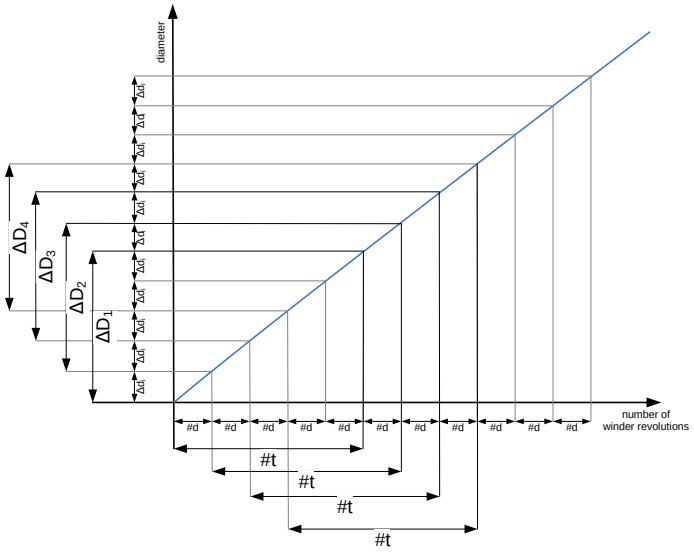
mt_i: Material thickness, calculated by FBWinder

#d: Number of winder revolutions of the thickness or diameter calculation

#t: Number of winding revolutions for the thickness calculation by the FBThicknessAverage block

k: Ratio #t / #d

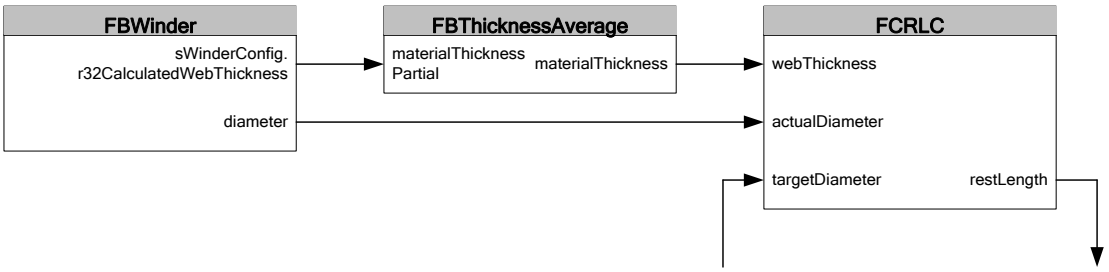
Figure 5-4



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Graphic representation of the function

The following diagram represents the logical connection with FBWinder. The trigger to accept the value can be generated from the value change of the variables latchDiamCalc or sWinderConfig.r32CalculatedWebThickness.



Warnings

Warnings are indicated by the status of the outputs *error* and *errorID*: *error* = FALSE and *errorID* <> 16#0000_0000.

Table 5-13 Warnings

ErrorID	Description
16#0000_0000	No warning
16#0209_4002	Array size is limited, increase *diameterMeasuringRevCount* AND/OR decrease *thicknessMeasuringCount* AND/OR Change LCONLIB_THICKNESS_MAX_SAMPLES constant value.

Errors

Error messages are indicated by the status of the outputs *error* and *errorID*: *error* = TRUE and *errorID* <> 16#0000_0000.

Table 5-14 Error messages

ErrorID	Description
16#0000_0000	No error.
16#0209_8005	Invalid configuration data: *thicknessMeasuringRevCount < diameterMeasuringRevCount* *or* *diameterMeasuringRevCount <= 0.0*

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

5.8 FBSetUnits – Setting conversion factors

Block name

FBSetUnits

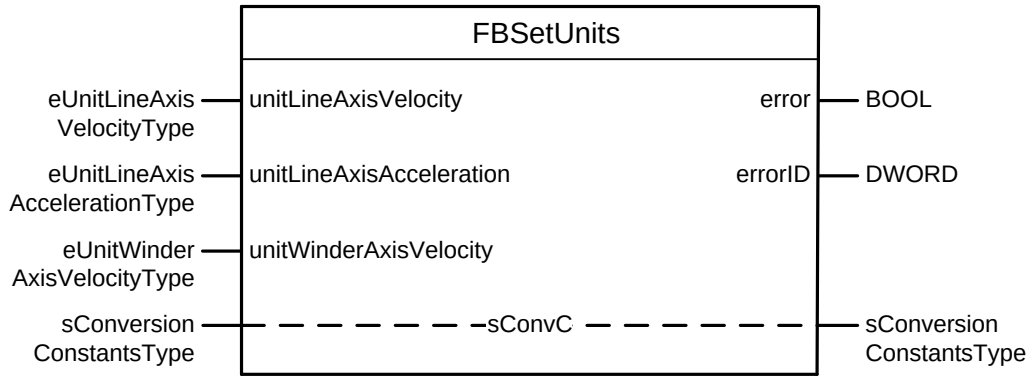
Task

Sets the internal conversion factors corresponding to the parameterized axis configuration.

Integration in the run-time system

The function block can be called in any task.

Graphic representation of the block



Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Block parameters

Input parameters

The input parameters are used to specify the axis units according to the selected TO parameterization.

Name	Data type	Initial value	Description
unitLineAxis Velocity	eUnitLine Axis VelocityType	NOT_ DEFINED	Axis configuration of the velocity corresponding to TO *lineAxis* NOT_DEFINED M_MIN: m/min MM_MIN: mm/min M_S: m/s MM_S: mm/s FT_MIN: ft/min INCH_MIN: inch/min FT_S: ft/s INCH_S: inch/s DEG_S_PRINT 360°/s = 1m/s

Converting Library

Name	Data type	Initial value	Description
unitLineAxis Acceleration	eUnitLine Axis Acceleration Type	NOT_ DEFINED	Axis configuration of the acceleration corresponding to TO *lineAxis* NOT_DEFINED M_S2: m/s² MM_S2: mm/s² FT_S2: ft/s² INCH_S2: inch/s² DEG_S2_PRINT 360°/s² = 1m/s²
unitWinderAxis Velocity	eUnitWinder Axis VelocityType	NOT_ DEFINED	Axis configuration of the velocity corresponding to TO *winderAxis* NOT_DEFINED RPM: U/min RPS: U/s DEG_MIN: °/min DEG_S: °/s

Input/Output parameter

The parameters included in this structure are used as output parameters and are automatically assigned by the system.

Name	Data type	Initial value	Description
sConvC	sConversion Constants Type		Structure with conversion factors Details on this structure are provided in the description of the type definitions.

Output Parameter

Name	Data type	Initial value	Description
Error	BOOL	FALSE	FALSE: No error TRUE: An error has occurred, refer to error identification In case of error (error = TRUE) the output tensionTaper is set to 1.0.
errorID	DWORD	16#0000_0000	Error identification, refer to error messages errorID <= 16#000_8000: warning errorID >= 16#0000_8000: error

Functionality

Using the input parameters, the velocity and the acceleration of the leading axis (*lineAxis*) as well as the velocity of the winder axis (*winderAxis*) are specified corresponding to the particular TO parameterization. This is realized using three enumeration types; these are orientated to the units of the physical quantity – metric and Anglo-American (imperial) units can be selected. The conversion factors of structure *sConvC* to convert into the required units, are automatically assigned by the system starting from the FB parameterization. This means that the user does not have to change them. Parameterization for printing machines is available with the DEG_Sx_PRINT option. This is orientated to the requirement that 360°/s = corresponds to 1m/s.

If one of the input parameters is incorrect or is not available, an error (*error*) is generated including an error number (*errorID*).

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Converting Library

Note

[LU] Length unit, e.g.: m, ft etc.
[LUb] Basis length unit, m or ft
[TU] Time units of the particular axis (under certain circumstances, different for lineAxis and winderAxis)

Errors

Error messages are indicated by the status of the outputs *error* and *errorID*: *error* = TRUE and *errorID* <> 16#0000_0000.

ErrorID	Description		
16#0000_0000	No error		
16#0206_8005	The requested dimension unit has not been implemented. Instead, standard conversion factors are used.		
	Input parameter	**Leading axis LU is in meters**	**Leading axis LU is in feet**
	unitLineAxisVelocity	M_MIN	FT_MIN
	unitLineAxisAcceleration	M_S2	FT_S2
	unitWinderAxisVelocity	RPM	RPM

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

Appendix

6 General information on the application

6.1 Scope of supply

The package Converting Library comprising:

- SIMOTION library as XML-Export
- Documentation

6.2 Changes/author

Table 6-1: Changes/author

Version	Date/change	Author
1.0	01.10.2006	Freddy Müller
1.1	01.05.2008	Freddy Müller
1.1.1.	05.06.2008 /user-defined units of measurement	Andreas Werner
1.2.	29.07.2009 New blocks, simplifications	Peter Tabori
1.2.1	23.07.2010 New blocks Simplifications Changelog in the library	Peter Tabori
1.2.2	23.09.2011 Simplifications Changelog in the library	Peter Tabori
1.2.3	26.01.2012 Bugfix in FCTaskIdentification Changelog in the library	Peter Tabori
1.3.0	02.04.2013 New winder control mode “Constant v Control”	Thomas Richter
2.0.0	06.08.2015 New Function Block FBCalcTargetDiamStopCondition; FBWinder description removed; Update FBDiameterCalculation;	Thomas Richter

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

7 Glossary

Table 7-1

Abbreviation	Meaning
FB	Function block
FUP	Function diagram
IN	Input parameter
IN/OUT	Input/output parameter
RFG	Ramp-function generator
LAD	Ladder diagram
LU	Length unit
NRM	Normalization value
MCC	Motion Control Chart (graphic programming language)
OUT	Output parameter
SCOUT	SIMOTION Controlling with optimized Usability Toolbox (Engineering System for the family of SIMOTION products)
ST	Structured Text (text-based programming language)
TO	Technology object
TU	Time unit
WHKL	Tension taper characteristic

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc

8 References

This list is in no way complete and only reflects a selection of suitable references.

Table 8-1

	Subject area	Title
/1/	Winder	Standard winder application with SIMOTION
/2/	Winder	Axial winder for the T400 technology module
/3/		T400, FM458-1, Simadyn D, SIMATIC TDC Function blocks

9 Contact partner

Application Center

SIEMENS

Siemens AG
Digital Factory
DF FA PMA APC
Frauenauracher Str. 80
91056 Erlangen
Germany
Fax: +49 (0)9131 98-1297
mailto: tech.team.motioncontrol@siemens.com

Copyright © Siemens AG 2015 All rights reserved
Manual_Converting_Library_V2.0.0_2015-08-06.doc